View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Digital Repository @ Iowa State University

IOWA STATE UNIVERSITY
Digital Repository

Graduate Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2016

Ab initio nuclear structure calculations with GPU acceleration

Hugh Potter
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

 Part of the Physics Commons

Recommended Citation

Potter, Hugh, "Ab initio nuclear structure calculations with GPU acceleration" (2016). *Graduate Theses and Dissertations*. 15793.
https://lib.dr.iastate.edu/etd/15793

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Ab initio nuclear structure calculations with GPU acceleration

by

Hugh Potter

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Nuclear Physics

Program of Study Committee:

James P. Vary, Major Professor

Pieter Maris

John G. Lajoie

Simanta Mitra

Soeren A. Prell

Iowa State University

Ames, Iowa

2016

Copyright © Hugh Potter, 2016. All rights reserved.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my advisor, Dr. James P. Vary, and my co-advisor, Dr. Pieter Maris, for their keen insight and unfailing support as I completed this work. I'd like to thank the rest of my thesis committee for helping to make my progress possible. Finally, I'd like to thank my family for supporting me both emotionally and materially in my aspirations, and my girlfriend Natalie for staying by me through the long haul.

This work was supported in part by the National Science Foundation under Grant No. PHY–0904782, the U.S. Department of Energy (DOE) under Grant Nos. DE-FG02-87ER40371 and DESC0008485 (SciDAC-3/NUCLEI). A portion of the computational resources were provided by the National Energy Research Scientific Computing Center (NERSC), a DOE Office of Science User Facility supported by the US DOE Office of Science under Contract No. DE-AC02-05CH11231, and by an INCITE award, "Nuclear Structure and Nuclear Reactions", from the US DOE Office of Advanced Scientific Computing. This research also used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the US DOE Office of Science under Contract No. DE-AC05-00OR22725.

ABSTRACT

We present a description of our *ab initio* nuclear structure formalism and its implementation in our highly parallel codebase. We discuss our strategy for incorporating hardware acceleration with Graphics Processing Units and the timing improvements that it produced. Finally, we use these systems to perform an extensive range of *ab initio* calculations for neutron drops in 10 and 20 MeV external harmonic-oscillator traps using chiral nucleon-nucleon plus three-nucleon interactions. We present ground state energies and energy differences, radii, internal energies, and level splitting physics results for neutron numbers $N = 2-40$ using the no-core full configuration model. We compare with quantum Monte Carlo results using AV8′ with Urbana IX and Illinois-7 3N forces, where available, and with the nonlocal NN interaction JISP16. These results lead to insights on the distinctions between candidates for the nuclear strong interaction and inform applications to develop new generations of energy density functionals for nuclei. In addition, in light of a promising new correlation between neutron drop and nuclear observables, we present expanded $N = 20$ neutron drop results and use the correlation to compare them with the experimental result for the difference in neutron and proton radii of ^{48}Ca.

CHAPTER 1. INTRODUCTION

1.1 Introduction

Over the past decade, there has been a growing interest in theoretical derivations and numerical simulations of the properties of nuclei using first principles, or "ab initio" methods. There is a major trend, for example, of employing nucleon-nucleon (NN) interactions that fit the NN phase shifts and deuteron properties, along with three-nucleon (3N) interactions tuned to properties of 3N systems. This work aims to build on that pathway by solving for the properties of exotic neutron systems and comparing those results with results from the neutron-rich nucleus ^{48}Ca.

In particular, there has been significant interest in *ab initio* solutions for systems of neutron drops trapped in external potentials aimed at providing insights into properties of neutron-rich nuclei and neutron star matter (1; 2; 3; 4; 5). Furthermore, a recent theoretical study shows a strong correlation between radii of neutron drops in harmonic oscillator traps and the thickness of neutron skins in ^{48}Ca and ^{208}Pb (6), which leads to comparisons with experimental data on these nuclei.

At the same time, comparisons of neutron drop results using different microscopic interactions provide information on the isovector part of the nucleon-nucleon (NN) interaction and the $T = 3/2$ component of the three-nucleon (3N) interaction. With these goals in mind, we extend the work of (5) with *ab initio* results for pure neutron systems through 40 neutrons in 10 and 20 MeV external harmonic oscillator (HO) traps. We use chiral NN+3N Hamiltonians and compare with results previously obtained using high-precision phenomenological NN+3N Hamiltonians.

We adopt no-core configuration interaction methods (e.g., see Refs. (7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17)), which have advanced rapidly in recent years, making it feasible to accurately solve fundamental problems in nuclear structure and reaction physics. We follow Refs. (2; 4) for the configuration interaction approach to trapped neutron drops in the current application.

At the same time, significant theoretical advances regarding the underlying Hamiltonians, constructed within chiral effective field theory (EFT) with ties to QCD, provide a firm foundation for nuclear many-body calculations (18; 19), leading us to adopt a chiral EFT Hamiltonian here. We also make use of the similarity renormalization group (SRG) approach (20; 21; 22; 23; 24; 25) which provides a straightforward and flexible framework for consistently evolving (softening) the Hamiltonian and other operators, including 3N interactions (14; 26; 27).

We provide results for neutron drop systems in 10 and 20 MeV HO traps using realistic chiral NN+3N interactions with uncertainty estimates where feasible. We present comparisons between our results and those of other high-quality NN+3N interactions. In particular, we compare with results obtained using the Green's function Monte Carlo (GFMC) and auxiliary field diffusion Monte Carlo (AFDMC) quantum Monte Carlo (QMC) methods (3; 4), where the Argonne v_8' (AV8′) NN interaction (28) was used in conjunction with the Urbana IX (UIX) 3N interaction (28) and with the Illinois-7 (IL7) 3N interaction (29). In selected cases, we also compare our results with those obtained using JISP16 (30; 31), a nonlocal NN potential without 3N interactions. The neutron drop results with JISP16 have appeared previously in Ref. (4).

We limit our main investigations to a single form of the chiral NN+3N interaction. That is, we use the chiral NN interaction at N^{3}LO with 500 MeV/c cutoff from Ref. (32) together with the chiral 3N interaction at N^{2}LO (33) in the local form of Ref. (34) for 500 MeV/c cutoff with low-energy constants determined entirely in the three-nucleon sector (35). This is also the Hamiltonian used in Refs. (14; 26; 27; 36; 37; 38) for *ab initio* studies of nuclear properties. We evolve this Hamiltonian using the free-space SRG to improve convergence. As in the earlier applications, we retain the induced many-body interactions up to the three-nucleon level and neglect induced four-nucleon (and beyond) interactions.

In addition, we present expanded neutron drop radius results at $N = 20$ in the context of a recent theoretical study of the correlation between neutron skin thickness in ^{48}Ca and ^{208}Pb and neutron drop radii (6). We present results at several combinations of SRG strength and UV cutoff, and use the correlation to compare with the values inferred from experiment.

Because of the extreme size of our calculations, we rely heavily on massively parallel supercomputers to perform our work. Driven by consumer demand for entertainment devices with powerful graphics capabilities, technology companies are investing large amounts of resources into improving dedicated graphics processing units (GPUs), which pack many computing units onto a single chip. Supercomputers like Titan and the upcoming Summit at Oak Ridge National Laboratory leverage this technology for scientic computing. We designed, implemented, and tested algorithmic improvements to our *ab initio* nuclear structure code that take advantage of these powerful computing resources.

We present this work in three parts. First, in chapters 2 and 3, we describe the theoretical background of our *ab initio* approach and the high-performance parallel code that implements it for both full nucleus and neutron drop calculations. Then, in chapter 4, we present a strategy for accelerating our code on Titan's GPUs and discuss the speedup we achieved with it. Finally, in chapters 5 and 6, we present the neutron drop physics results we have calculated for a wide range of systems in 10 MeV and 20 MeV external harmonic traps, and our ^{48}Ca results and comparison to neutron drop systems.

1.2 Technology

We perform our calculations with the parallel eigensolver Many-Fermion Dynamics for nuclei (MFDn) (39; 40; 41; 42; 43; 44; 45; 46). MFDn is written in FORTRAN, and uses the Message Passing Interface (MPI) and the Open Multiprocessing (OpenMP) library for parallelism. See chapter 3 for a discussion of MFDn and these technologies.

Except where otherwise specified, we present results from the DOE supercomputer Titan at the Oak Ridge National Laboratory. Titan is a Cray XK7 supercomputer with 18,688 physical compute nodes, each of which has one 16-core 2.2 GHz AMD Opteron 6274 processor and 32 GB of RAM. Each node is divided into two NUMA domains, and nodes are served in groups

of two by Gemini high-speed interconnect routers. Additionally, each node has one NVIDIA K20 Kepler GPU accelerator with 6 GB of memory.

CHAPTER 2. THEORETICAL APPROACH

2.1 Overview

We adopt the No-Core Full Configuration (NCFC) method for our calculations, following Refs. (10; 7; 8; 9). We take precalculated 2- and 3-body nuclear potentials in a harmonic oscillator (HO) basis and use them to generate a full A-body Hamiltonian matrix in a truncated HO basis for our nuclear system. See section 2.2 for a more detailed discussion of the basis. Unlike similar methods, where filled shells or sub-shells of a mean field are treated as a single-particle "core" that interacts with valence nucleons, we simulate all nucleons as individual particles with the same standing in the calculation.

We discuss our basis in Section 2.2, and in Section 2.3 we discuss the components of our nuclear and neutron-drop Hamiltonians. In Section 2.4 we discuss our choice of 2- and 3-body inter-nucleon potentials.

2.2 Basis

We represent our basis states as combinations of single-particle HO eigenstates, each of which has the following definite quantum numbers: radial quantum number n, orbital angular momentum l, spin $s = 1/2$, total angular momentum j with z-projection m_j, and total isospin $T = 1/2$ with z-projections $t = \pm 1/2$. Nucleon total spin and total isospin are always the same, so we do not track them. We work only in bases of good single-particle j, so henceforth we will use m and m_j interchangeably. We can represent a single particle state (SPS), then, as $|nljmt\rangle$. We refer to this as an m-scheme basis because of definite m_j; see section 4.2 for an alternative approach. The choice of an HO basis introduces an HO parameter $\hbar\omega$. We use

a lowercase ω to distinguish between basis parameter $\hbar\omega$ and the strength $\hbar\Omega$ of the external HO potential that we impose for neutron-only systems.

We combine these SPSs into many-body states (MBSs) by constructing Slater determinants. A Slater determinant is an antisymmetrized linear combination of SPS products of the following form:

$$\Phi(r_1, r_2, \ldots, r_A) = \frac{1}{\sqrt{A!}} \begin{vmatrix} \phi_1(r_1) & \phi_2(r_1) & \ldots & \phi_A(r_1) \\ \phi_1(r_2) & \phi_2(r_2) & \ldots & \phi_A(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(r_A) & \phi_2(r_A) & \ldots & \phi_A(r_A) \end{vmatrix} \tag{2.1}$$

Equation 2.1 shows the position-space representation of an MBS for clarity in indexing, but, as described above, we work in an HO basis. Switching two columns in a matrix preserves the magnitude of the derminant but reverses its sign, so a Slater determinant MBS has the correct fermionic behavior under coordinate switches.

The full basis space is infinite, so we must also choose a truncation method. We introduce another parameter $N_{\max}$, defined in terms of the sum of the SPS HO quanta N_i. $N_{\max}$ is the number of HO quanta above the minimal number of quanta N_0 required to satisfy the Pauli exclusion principle for the A nucleons of the nuclear system in question:

$$N_{\max} = \sum_{i=1}^{A} N_i - N_0 = \sum_{i=1}^{A} (2n_i + l_i) - N_0 \tag{2.2}$$

Our basis, then, is parameterized by $N_{\max}$ and $\hbar\omega$.

For a given nuclear system, we perform runs over a range of $N_{\max}$ and $\hbar\omega$ values, producing a "convergence pattern" of results. Ideally, we achieve convergence by extending our calculations to a region of $N_{\max}$ and $\hbar\omega$ space where our results do not depend on $N_{\max}$ and $\hbar\omega$; we can then take results in that region to be final. Basis size grows extremely quickly with $N_{\max}$, however, so it is often not feasible to converge our results. In these cases, we use several methods to extract a result from the convergence pattern; see section 5.3 for a discussion of energy extrapolation and section 5.5 for a discussion of calculating radii.

2.3 Many-Body Hamiltonian

We now present the many-body Hamiltonian for a full nuclear system with protons and neutrons, constructed from the kinetic energy term and contributions from our input 2- and 3-body inter-nucleon potentials V^2 and V^3. To preserve the translational invariance of the nuclear Hamiltonian, we use the relative form of the kinetic energy operator:

$$H^A = T_{\rm rel} + V = \frac{1}{2M}\sum_{i<j}(p_i - p_j)^2 + \sum_{i<j} V^2_{ij} + \sum_{i<j<k} V^3_{ijk} \tag{2.3}$$

where M is the total mass of the nuclear system.

Our Hamiltonian operator is translationally invariant, but our single-particle states, as described in Section 2.2, use absolute coordinates. This gives the Hamiltonian described in Equation 2.3 a number of eigenstates with spurious excitations of the center-of-mass (COM) of the system, which we wish to remove from our results. Because we work in a harmonic oscillator basis truncated by $N_{\rm max}$ as described in Section 2.2, we can factor our nuclear wavefunctions exactly into a pure harmonic oscillator wavefunction that depends only on the COM motion and a translationally-invariant wavefunction that depends only on relative coordinates.

To exploit this factorization, we introduce a Lagrange multiplier term as in (47). For convenience, we define an operator $H_{\rm cm}$, the COM Hamiltonian, with an additional harmonic potential term $U_{\rm cm}$ of the same strength $\hbar\omega$ as our basis:

$$H_{\rm cm} = T_{\rm cm} + U_{\rm cm} = \frac{1}{2M}\left(\sum_i p_i\right)^2 + \frac{1}{2}M\omega^2(\sum_i r_i)^2 \tag{2.4}$$

Our Lagrange multiplier term is then $\lambda(H_{\rm cm} - \frac{3}{2}\hbar\omega)$. The contribution of this term is zero for states with COM motion restricted to the lowest HO state, the $0s$ state; for states with COM excitations, however, it lifts their energy levels by $\lambda\hbar\omega$ per COM excitation. Our diagonalization routine (see section 3.5) produces the lowest n states for our choice of n, which is usually around 5 – 10. If the excitation energy of the highest excitation state that we calculate is below $\lambda\hbar\omega$, then, we are assured that all our calculated states are free of spurious COM excitations. Our many-body solutions of interest for comparison with experiment, then, are these low-lying states, all of which are guaranteed to have pure $0s$ COM motion, which we can accomodate

when calculating observables. With the Lagrange multiplier term, our final nuclear many-body Hamiltonian $\tilde{H}^A$ is

$$\tilde{H}^A = T_{\text{rel}} + \sum_{i<j} V_{ij}^2 + \sum_{i<j<k} V_{ijk}^3 + \lambda(H_{\text{cm}} - \frac{3}{2}\hbar\omega) \tag{2.5}$$

We move now to pure-neutron systems, which are not self-bound, and require an external potential to bind them together. We select an external HO potential for this work. This anchors the system to an absolute reference point in position space, so we no longer need to use the relative form of the kinetic energy operator or separate out the COM motion. We build the neutron drop many-body Hamiltonian H^N, then, as

$$H^N = \sum_i \frac{p_i^2}{2m} + \frac{1}{2}m\Omega^2 \sum_i r_i^2 + \sum_{i<j} V_{ij}^2 + \sum_{i<j<k} V_{ijk}^3 \tag{2.6}$$

where Ω paramaterizes the external harmonic potential. For details of how we construct the corresponding many-body Hamiltonian matrices programmatically, see section 3.4.

2.4 Chiral Interactions

As discussed in section 2.1, one input to our calculation is the inter-nucleon potential, which we read in as two- and three-body nucleon-nucleon potential matrices. We use potentials derived from chiral perturbation theory (χPT), an effective field theory for low-energy quantum chromodynamics (QCD). The underlying symmetries of QCD, and in particular chiral symmetry, serve as a theoretical foundation for χPT and a theoretical link between the two theories. In χPT, nucleons and pions are the effective degrees of freedom; we write down the most general Lagrangian consistent with the QCD symmetries, and use that to generate all interaction graphs through a specified order. We can then use these graphs to calculate inter-nucleon chiral potentials. To suppress the characteristic non-perturbative high-momentum regions of the strong force, we use a momentum-dependent "regulator" function parameterized by a momentum "cutoff" Λ. We use chiral interactions with $\Lambda = 500$ MeV/c at both the two- and three-body level, calculated with graphs through N^{3}LO at the two-body level (32) and N^{2}LO at the three-body level (33).

We use input potentials evolved by similarity renormalization group (SRG) techniques with the methods of Ref. (27); we base the following discussion on that source. Our SRG transformation is a unitary transformation that softens the coupling in the Hamiltonian between states of differing momentum. We parameterize this unitary transformation by the "flow parameter" α, which controls the degree of softening:

$$H_\alpha = U_\alpha^\dagger H U_\alpha \tag{2.7}$$

We are interested in the effect that varying α has on H_α; that is, the derivative of H_α with respect to α. It is convenient to introduce an antihermitian operator η_α, defined by

$$\frac{d}{d\alpha}U_\alpha = -U_\alpha \eta_\alpha \tag{2.8}$$

so that

$$\frac{d}{d\alpha}H_\alpha = [\eta_\alpha, H_\alpha] \tag{2.9}$$

Our SRG transformation, then, is defined by our choice of the "generator" η_α. Following Ref. (27), we choose

$$\eta_\alpha = (2\mu)^2 [T_{\rm rel}, H_\alpha] \tag{2.10}$$

where μ is the reduced nucleon mass and $T_{\rm rel}$ is the relative kinetic energy defined in section 2.1. Note that this choice of η_α vanishes when H_α commutes with $T_{\rm rel}$. Equivalently, η_α vanishes when H_α is diagonal in the basis of eigenstates of $T_{\rm rel}$. As we increase α, then, we expect H_α to approach this stationary point. Since $T_{\rm rel}$ is essentially a momentum operator, its eigenstates are momentum eigenstates. A matrix that is diagonal in this momentum eigenbasis is momentum-decoupled: i.e., does not possess couplings between different momentum eigenstates. We can then adjust α to tune the degree to which we soften the momentum coupling. Unless otherwise specified, the results presented in this work use input potentials transformed with flow parameter $\alpha = 0.08$ fm^4.

Unfortunately, this transformation induces higher-body terms; a pure 2- and 3- body interaction, transformed by an SRG transformation as described above, gains contributions at the four-body level and all higher levels. NCFC calculations with 4-body forces and higher are infeasible with present computational resources and techniques, so we neglect the induced

higher-body terms. This sacrifices our guarantee of unitarity, so we must verify that we produce the same results to a suitable accuracy with and without the SRG transformations we employ. For smaller systems, the calculations may be performed both with and without this transformation to the accuracy needed to assess the importance of the induced higher-body interactions. Alternatively, we can test the dependence of results on the flow parameter itself to gauge the role of the neglected higher-body terms. For the neutron drop case, we find that we have substantial freedom in the choice of flow parameter. In the study presented in Ref. (5), we find little difference between neutron drop results with $\alpha = 0.04$ fm^4 and $\alpha = 0.08$ fm^4; see section 5.2 for further discussion.

CHAPTER 3. MFDn

3.1 Overview

Calculations of the magnitude discussed in this work require the use of supercomputers and specialized parallel codes to take advantage of them. We use MFDn, a hybrid MPI-OpenMP eigensolver written in Fortran (39; 40; 41; 42; 43; 44; 45; 46). MFDn uses MPI for inter-process communication and OpenMP for threading.

3.2 MPI and OpenMP

MPI, the Message Passing Interface, facilitates communication between different processing elements. An individual execution context in MPI is called an "MPI process"; processes exist in their own memory spaces, and communicate through MPI functions. Processes post matching send and receive requests; when matching requests are posted, MPI coordinates information transfer between them. MPI is written to be implemented as a library, callable from C or Fortran.

OpenMP is a standard for managing multiple parallel threads in the same memory space. Instead of explicitly calling communication functions and managing data, programmers using OpenMP insert compiler directives to parallelize regions of code. The compiler then fills in the implementation details, allowing for faster development times. OpenMP's syntax is developed around a shared memory model of communication, wherein all threads have access to the same block of RAM, so it is usually unsuitable for communication between different physical units of computation.

One common practice is to use MPI for coarse-grained parallelism between different physical nodes, and then use OpenMP for fine-grained parallelism between threads on a node. On Titan,

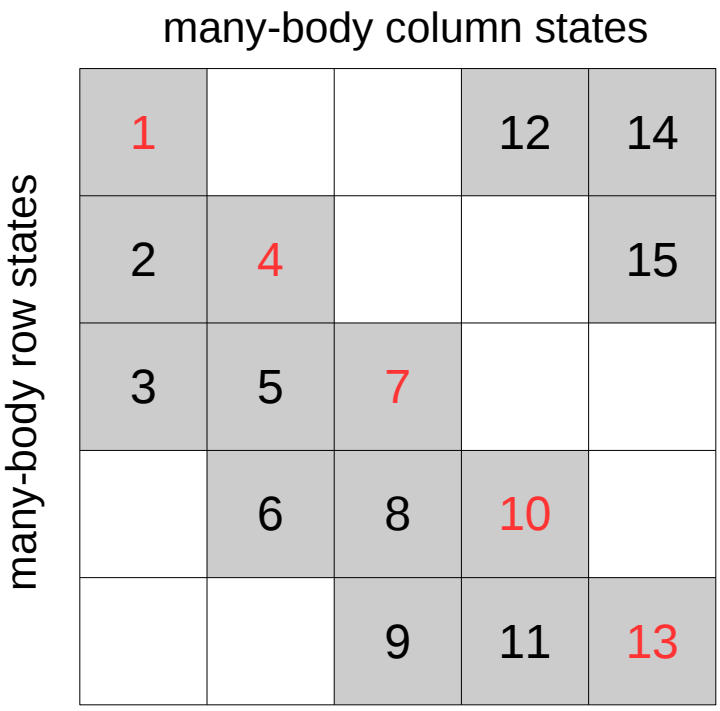

Figure 3.1 (Color online.) Square grid arrangement of 15 MPI processes in MFDn; diagonal processes (1, 4, 7, 10, 13) are marked in red. We exploit the symmetry of the Hamiltonian by storing only half of the off-diagonal blocks. The three empty blocks in the lower left are stored in the upper right. This arrangement ensures that each row and column of the grid has the same number of MPI processes.

we use two MPI processes per physical node. Nodes on Titan are divided into Non-Uniform Memory Access (NUMA) domains; RAM is split across NUMA domains, and memory accesses are optimized between processors and RAM in the same NUMA domain. We use two MPI processes per node, or one MPI process per NUMA domain, to avoid cross-NUMA-domain memory accesses.

MPI processes in MFDn are arranged to communicate in a square grid topology to facilitate organization and communication for matrix operations, as illustrated in Fig. 3.1. Each MPI process is indexed by a row and a column, and we identify a set of diagonal processes with equal row and column index, marked in red in Fig. 3.1. The many-body Hamiltonian is symmetric, so, with some exceptions for convenience on the diagonals, we save memory by storing only half the matrix elements and filling only half the grid topology with processes.

3.3 Basis Generation

In the setup phase, MFDn generates the many-body basis and distributes it across all MPI processes. We store SPSs as flat integer indices, and MBSs as arrays of those indices. For a given set of A SPSs, we store only one MBS permutation. To avoid duplication, we generate only MBSs where the integer SPS indices increase from left to right.

We generate the many-body basis on diagonal processes only. For the matrix construction stage, discussed in section 3.4 below, it is convenient to divide the full Hamiltonian into many smaller "tiles"; we can then use a coarse-grained check to determine if the entire tile contains only zeroes and can be ignored. With this in mind, we divide the basis into groups of MBSs based on SPS n, l, and j. To keep group size under control for larger nuclear systems, we partition into groups based on SPS m as well, giving each group a small range of m values.

As the diagonal processes iterate through many-body basis groups, they assign groups to themselves according to a round-robin scheme. Each diagonal process then broadcasts its basis groups both to all processes in the same row and to all processes in the same column. As shown in Fig. 3.1, we arrange processes so that each row and each column have the same number of processes. After the basis setup, then, each off-diagonal process contains two sets of basis groups, one received from the diagonal process in the same column and one received from the diagonal process in the same row. We refer to these as the row groups and the column groups; diagonal processes contain only one set of groups, which they use as both the row and column groups. The submatrix assigned to a process is defined by the MBSs in its row and column groups.

3.4 Matrix Generation Phase

MFDn generates the full many-body Hamiltonian during the matrix generation phase. We require several input matrices for this process; for neutron drop calculations, we read in the 2- and 3-body input interactions from files and store them in memory. As discussed in section 2.1, for nuclear calculations we require relative kinetic energy and squared-radius operators; these are not trivial to calculate, so we store these in files and read them in at runtime as well.

All input matrices are stored once per MPI process. We store our input matrices in-core in an efficient angular momentum and isospin-coupled basis, and access them with a GPU-accerated decoupling routine; we discuss the details of the coupled basis and the decoupling transformation in chapter 4 and the appendix.

An element in the full many-body Hamiltonian matrix is specified by its row and column MBSs. If more than three SPSs in the row and column MBSs differ, three-body potentials do not contribute to that many-body matrix element. In the matrix generation stage, then, MFDn must locate and calculate all matrix elements with row and column MBSs differing by three or less SPSs. The first step is a comparison of the quantum numbers for the row and column groups, as described in Section 3.3. Many tiles can be rejected immediately as containing only zeroes. Tiles with potential nonzero elements are generated in one of two ways, depending on how similar or dissimilar the row and column group quantum numbers are.

For similar quantum numbers, we expect a tile to contain more nonzeroes. In these cases we iterate through all possible matrix elements in the tile and check each to see if each must be zero. For dissimilar quantum numbers, we expect a sparser tile; in those cases we iterate through all the quantum numbers that could produce nonzero elements. MFDn runs through nonzero elements twice during the matrix generation phase: once to determine how many nonzeroes there are so the nonzero storage arrays can be allocated, and once to calculate the elements. For either method, we apply OpenMP parallelism in each process.

We build each matrix element as a sum of contributions from the input matrices. Elements where row and column state MBSs have three different SPSs, for example, include only one contribution from the input 3-body potential. That contribution is the input matrix element indexed by the three SPSs that are different in the row state and the three SPSs that are different in the column state. (Depending on the format in which we store the input potential, this contribution might itself be a sum over several stored matrix elements; see Section 4.2 for details.) For an element with only two differences, however, we have only two differing row and column SPSs; we must iterate over all "spectator" nucleons in the same row and column state, summing their contributions. For elements with only one difference we have a double sum over two nested iteration indices, and for diagonal elements we have a triple sum.

Algorithm 1 Lanczos Algorithm

procedure LANCZOS(A, m) ▷ diagonalize matrix A using m Lanczos iterations
  $v_0 \leftarrow 0$ ▷ the Lanczos vectors v_i are vectors dimension-compatible with A
  $v_1 \leftarrow$ RANDOMUNITVECTOR()
  $\beta_1 \leftarrow 0$
  for $i = 1$ **to** m **do**
    $w_j \leftarrow Av_j - \beta_j v_{j-1}$
    $\alpha_j \leftarrow w_j \cdot v_j$
    $w_j \leftarrow w_j - \alpha_j v_j$
    $\beta_{j+1} \leftarrow ||w_j||$
    $v_{j+1} \leftarrow w_j / \beta_{j+1}$
  end for
  $T =$ TRIDIAGONALMATRIX(α, β) ▷ α is along the diagonal, with β the off-diagonals
  $eigenvalues =$ EIGENVALUES(T)
  $eigenvectors = v^T T v$
  return $\{eigenvalues, eigenvectors\}$
end procedure

3.5 Diagonalization Phase

We use the Lanczos algorithm (48) to obtain a set of the lowest eigenvalues and eigenvectors from the generated many-body Hamiltonian. The Lanczos algorithm is a variant of the power method, a simple iterative routine in which we start with an initial random vector and multiply it many times by the matrix we seek to diagonalize. After many iterations, the vector converges to the eigenvector with the largest-magnitude eigenvalue.

The Lanczos algorithm makes use of the intermediate information produced by the power method to generate a symmetric, tridiagonal matrix with the same eigenvalues as the matrix we seek to diagonalize. See Algorithm 1 for a description of this process. The Lanczos algorithm can be written in several forms; in Algorithm 1, we follow the conventions of Ref. (49).

During the course of the calculation we generate a series of Lanczos vectors v_i, which we can use to transform the tridiagonal matrix into a matrix of the original matrix's eigenvectors. We recover only a small subset of the eigenvectors this way. The lowest-energy eigenvectors converge first; we periodically multiply them by the many-body Hamiltonian to check their convergence, and use that as a measure of when to stop the Lanczos iterations. Higher-energy eigenvectors, then, are not well-converged at the end of the calculation.

In addition to the base algorithm described in Algorithm 1, we employ an additional orthonormalization step; after each iteration, we orthonormalize the new Lanczos vector against each previous Lanczos vector to improve numerical stability. We use the LAPACK linear algebra library to calculate the eigenvalues of the tridiagonal matrix.

The subset of eigenvalues of the diagonalized matrix that we calculate are the energies of the lowest states of our nuclear system, and the eigenvectors represent the corresponding states in our harmonic oscillator basis. We process the eigenvectors to calculate other observables, including total angular momentum J, radius, internal energy (defined in Section 5.5), and various transition matrix elements and other quantities of interest. In particular, our many-body Hamiltonian is rotationally invariant, and shares an eigenbasis with J. This serves as a useful consistency check; we expect integer J for systems with an even nucleon count and half-integer J for odd systems. The eigenvectors are saved to disk for possible further calculation.

CHAPTER 4. GPU ACCELERATION

4.1 GPU Accelerators

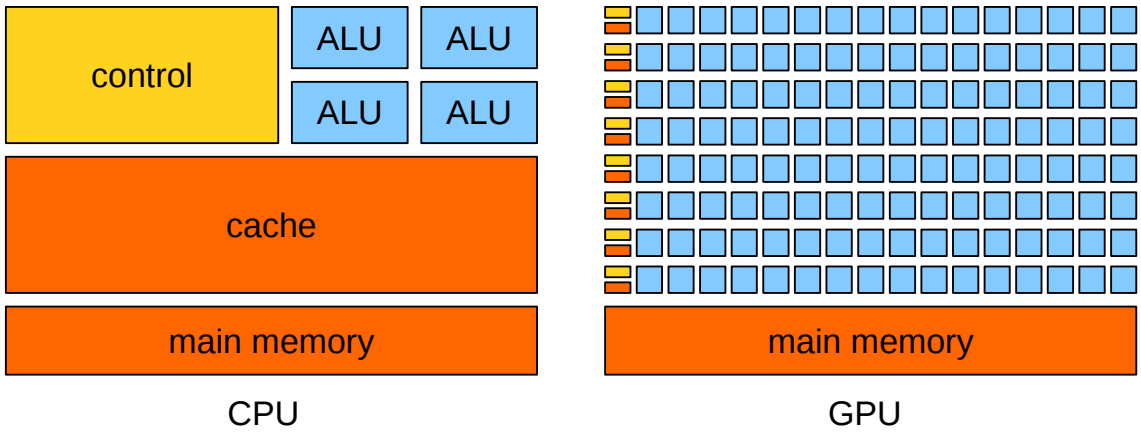

Figure 4.1 (Color online.) Schematic illustration of the differences between CPU and GPU computing; the CPU has several powerful calculating units (here denoted ALU, for algorithmic logic unit) with sophisticated caching and other optimizations, whereas the GPU has many less-powerful units. Figure adapted from Ref. (50).

Graphics Processing Units (GPU) are specialized hardware accelerators designed for highly parallel calculations that favor arithmetic over memory access. Broadly speaking, as illustrated in Fig. 4.1, a conventional CPU uses sophisticated prediction and caching systems to achieve exceptional speed with a limited number of powerful computational units, and a GPU sacrifices those caching and prediction systems in favor of a large number of less powerful computational units. GPUs can produce an extreme speedup when the task can be divided into a large number of independent tasks. GPUs generally have high memory bandwidth to and from the CPU-controlled main memory, but high memory latency; memory transfer requests to and from GPU memory have a delay before starting, but once started they are extremely fast. This means that very large transfers are efficient, but many small transfers are very inefficient.

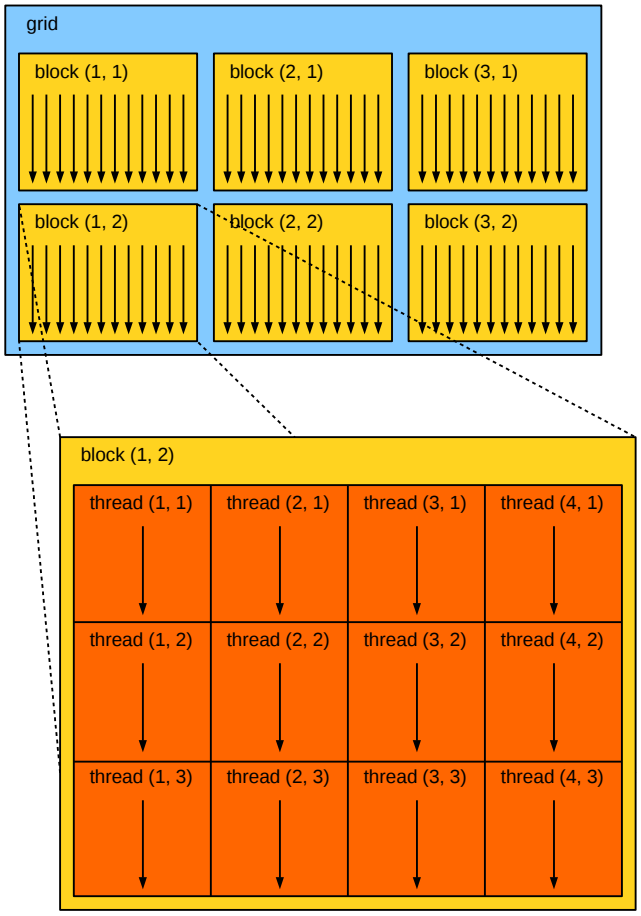

Figure 4.2 (Color online.) Hierarchy of parallelism in CUDA; a single GPU calculation runs on a grid of blocks; each block has a number of threads of execution. The user generally configures grid and block size to match the structure of the problem. Block and thread indices are illustrated in the figure; each thread can access its indices to determine which fragment of the overall calculation to perform. Figure adapted from Ref. (50).

We use NVIDIA's Compute Unified Device Architecture (CUDA), a parallel computing platform and programming model designed to interface with the GPUs (50). GPUs perform data-parallel "single-instruction, multiple data" (SIMD) calculations; that is, they execute the same instructions across many different pieces of data. In NVIDIA's and CUDA's model of GPU computation, a calculation is broken into a "grid" of "blocks", each of which consists of multiple threads of execution, as illustrated in Fig. 4.2. The GPU's computational resources are divided into "streaming multiprocessors", each of which can support multiple blocks with many simultaneous threads. Blocks are executed immediately or stored for later execution;

when a block finishes execution, one of the stored blocks takes its place, and so on until all blocks have finished.

Threads in CUDA are grouped into "warps" for execution; each warp consists of 32 threads. The warp executes the same instructions across all threads. When the code branches, as with conditional statements, the warp executes the different possibilities one-at-a-time. During such branches, a thread that does not follow the currently executing conditional branch is "inactive"; only "active" threads execute commands. GPU code therefore tends to be more efficient when it has fewer conditional branches, as that produces less thread inactivity.

CUDA allows us to write our GPU kernel in a version of the C language that has been slightly augmented to support GPU calculations. CUDA C code is divided between "host" code that runs on the CPU and "device" code that runs on the GPU. The device code defines a "kernel" that we call with a "kernel invocation" from the CPU code. When we perform a kernel invocation, we can specify a number of blocks of threads and a number of threads per block.

Inside a kernel, the programmer is responsible for mapping threads to pieces of work. CUDA defines variables in the kernel through which to access block and thread IDs; each thread uses those to determine which piece of the calculation to address. For example, in a program that adds two arrays A and B to form array C, one thread might inspect its block and thread IDs and calculate that it is responsible for element 342; that thread would then calculate $A_{342}+B_{342}$ and store the result in C_{342}.

In this example, A, B, and C are all in GPU memory. A complete CUDA-accelerated array addition program would have several steps. First, the CPU requests three blocks of memory on the GPU; CUDA returns pointers to GPU memory. Next, the CPU issues a command to copy A and B into two of the blocks of GPU memory. When those copies have completed, the CPU invokes the kernel, supplying pointers to the three blocks of GPU memory. The GPU threads run the kernel, filling the C block with sums. Finally, when the kernel finishes the CPU issues a command to copy the C block of GPU memory back into a block of CPU memory.

4.2 The Coupled-JT Format

As discussed in section 2.2, we operate in an m-scheme basis where each SPS has definite quantum numbers n, l, j, m, and t. A matrix element for a 3-body input interaction like the chiral interactions we use can be specified by two 3-body states, which we refer to as the row and column states. Each 3-body state has three SPSs, and each SPS has its own independent quantum numbers. This format is convenient for many-body calculations, but it is not sufficiently compact for our applications. We can achieve substantial memory savings by coupling the SPS angular momenta and isospin together and using the coupled values instead of the independent SPS z-projections to define the basis. In one representative case, a 3-body interaction is 33 gigabytes (GB) of RAM in the m-scheme basis, but only 1 GB in the coupled-JT basis (14; 27; 51). Since we must store input matrices once per MPI process, or twice per node on Titan (each node has 32 GB of memory), memory savings of that magnitude can be extremely important.

A 3-body m-scheme state has this form: $|n_a l_a j_a m_a t_a\rangle|n_b l_b j_b m_b t_b\rangle|n_c l_c j_c m_c t_c\rangle$. The coupled-$JT$ basis couples j_a with j_b into J_{ab} and J_{ab} with j_c into $J_{(ab)c}$, and likewise for isospin (as in Section 2.2, we use T for total isospin). Coupled-JT states, then, have the following form: $|n_a l_a j_a;\ n_b l_b j_b;\ n_c l_c j_c;\ J_{ab} J_{(ab)c} T_{ab} T_{(ab)c}\rangle$. See the appendix for a more detailed mathematical description of the coupled-JT basis.

Both the coupled-J and the m-scheme bases have good n, l, and j for each SPS, so we can divide up both many-body basis schemes into segments with different SPS nlj values. Because of their orthogonality, the projections between basis states in both many-body bases will be zero if the states have different nlj values. Similarly, we can divide matrices into nlj blocks; when performing basis transformations between matrices in the m-scheme and coupled-JT bases, an element in a matrix in one basis depends only on elements in the same nlj block of the matrix in the other basis; nlj blocks do not mix.

Because our input Hamiltonians are rotationally invariant, their matrix elements are zero where the row and column states have different total J. This is impossible to exploit in m-scheme, where total J is not a good quantum number; in a coupled-JT input Hamiltonian,

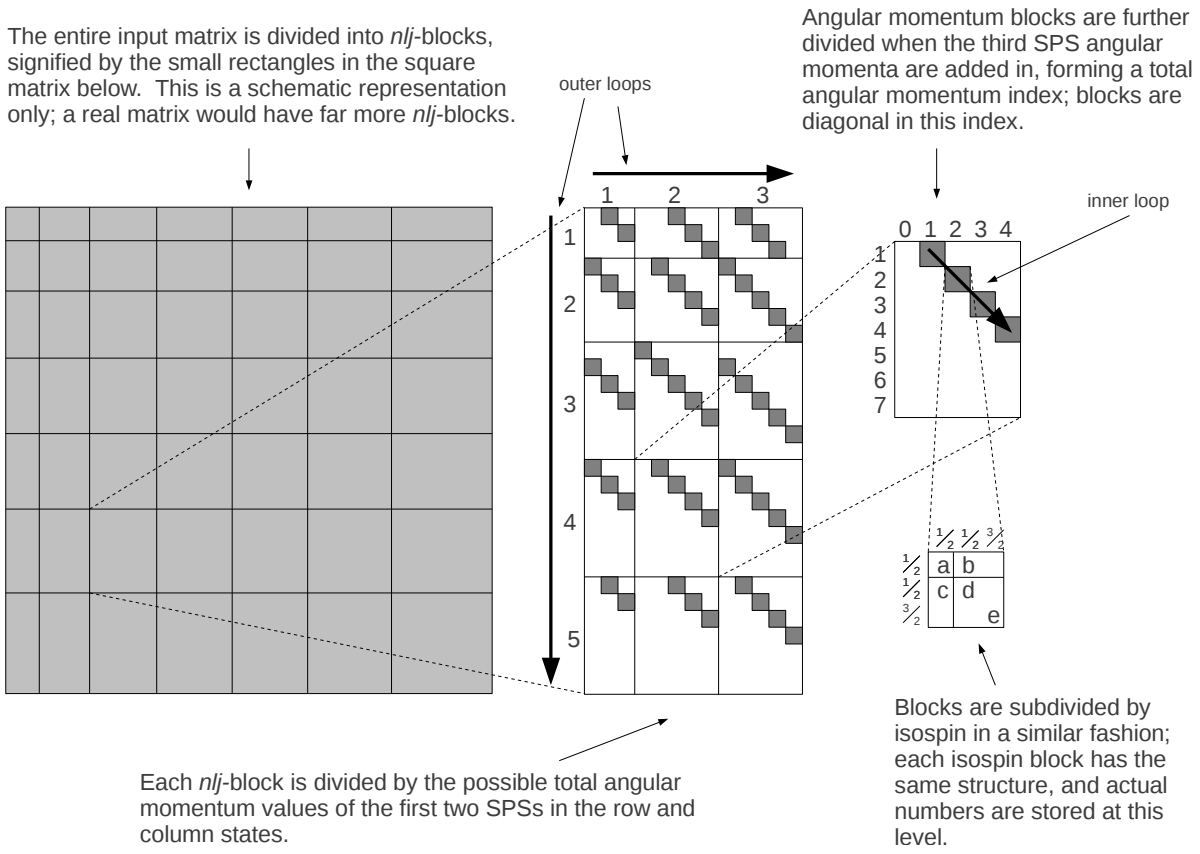

Figure 4.3 Structure of the input 3-body Hamiltonian in the coupled-JT basis. By using a basis with good total J and T for each MBS, we are able to exploit the rotational invariance of the Hamiltonian and produce a number of rectangular diagonal submatrices. Figure reproduced from Ref. (52).

however, we can restrict our attention to the elements that are diagonal in $J_{(ab)c}$ and disregard the others as zero. We say that matrices are diagonal in quantum number x when the row-state x and the column-state x are equal for all nonzero elements; we apply this terminology to square and non-square matrices alike. In practice, we organize our coupled-JT matrices first by nlj-block, then by J_{ab}, then by $J_{(ab)c}$, producing a number of blocks with the same nlj and J_{ab} that are diagonal in $J_{(ab)c}$. See Fig. 4.3 for a graphical representation of the structure of our coupled-JT matrices.

Similar considerations apply to isospin, though over a considerably smaller space. Each block with definite nlj, j_{ab}, and $j_{(ab)c}$ is further subdivided by coupled total isospins T_{ab} and $T_{(ab)c}$, as shown in Fig. 4.3. All SPSs have the same isospin magnitude $T = 1/2$, so the isospin structure is the same regardless of the angular momentum quantum numbers. T_{ab} is 0 or 1,

and $T_{(ab)c}$ is 1/2 for $T_{ab} = 0$ and 1/2 or 3/2 for $T_{ab} = 1$. Because isospin blocks are diagonal in $T_{(ab)c}$, each isospin block contains only 5 nonzero elements; we store numbers at this level.

4.3 Decoupling Algorithm

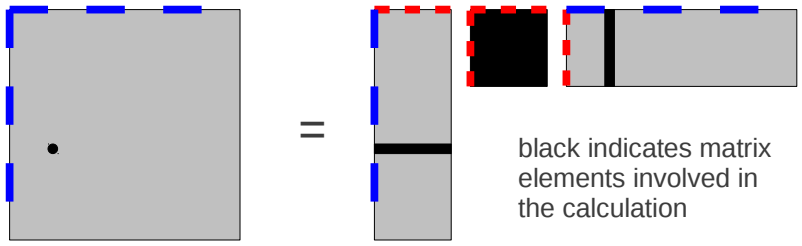

Figure 4.4 (Color online.) Basis conversion from a compact coupled-JT basis, shown in red, to an m-scheme basis, shown in blue. The basis conversion is equivalent to matrix multiplication, but for our usage we recover m-scheme elements one-at-a-time, as indicated by the black elements above. Figure reproduced from Ref. (52).

Before we can use matrices stored in the coupled-JT basis, we must decouple them back into m-scheme. We use a decoupling routine written in the C language. Schematically, the decoupling is a basis transformation, which can be written as a matrix multiplication:

$$V_m^3 = D^T V_{JT}^3 D \tag{4.1}$$

where V_{JT}^3 is the input 3-body Hamiltonian in the coupled-JT scheme as it is stored in memory, V_m^3 is the input 3-body Hamiltonian as it is used by MFDn, and D and D^T are basis transformation matrices built from projections between the m-scheme and coupled-JT bases. For our application, we do not generate or store the V_m^3, D, and D^T matrices in their entirety, as they would usually exceed available memory; we store the V_{JT}^3 matrix and generate elements from V_m^3 on-the-fly as MFDn requests them. We also generate elements of D and D^T on-the-fly as they are needed for this transformation.

Fig. 4.4 shows a high-level illustration of a single H_m^3 element calculation. In principle an element of V_m^3 is a linear combination of all elements from V_{JT}^3, weighted by some elements from D and D^T. Many elements from V_{JT}^3 do not contribute, however, because of orthogonality relations that manifest as zeroes in V_{JT}^3 and D. This sparsity structure is highly predictable and can be exploited.

As discussed in section 4.2, only V^3_{JT} elements in the same nlj block as the requested V^3_m element contribute. The V^3_{JT} matrix is stored linearly in a block of memory, segmented first into nlj blocks. As part of the setup phase of our calculations, we precalculate a six-dimensional indexing array of indices into V^3_{JT} that maps row and column nlj to the start location of the corresponding block. Because of the symmetry of V^3_{JT}, which allows us to store only half of the matrix, the range of later indices in this indexing array depends on the value of earlier indices; in addition, we employ a truncation scheme for V^3_{JT} that depends on the sum of the SPS HO quanta, exacerbating this effect. As a result, the array is not a hyperrectangle in the six-dimensional space of its coordinates, and we refer to it as jagged or non-rectangular.

Because of the total angular momentum conservation, nlj blocks in V^3_{JT} are diagonal in total angular momentum, and we can express the iteration over an nlj block with three nested iterations. Henceforth, we refer to iteration over a set range as a for-loop. Recall from section 4.2 that we store coupled angular momenta J_{ab} and $J_{(ab)c}$ for a coupled-JT basis state. The first for-loop is over the coupled angular momentum J_{ab} for the row state, and the second is over J_{ab} for the column state. The third for-loop is over the coupled total angular momentum $J_{(ab)c}$ for the row and column states simultaneously; because states with different $J_{(ab)c}$ are orthogonal, we require only 1 for-loop here. Note that the range of iteration of the $J_{(ab)c}$ for-loop depends on the current iteration of the outer J_{ab} loops in which it is nested. Fig. 4.3 illustrates the sparsity structure of V^3_{JT} and how the three for-loops iterate over it.

We must also calculate and insert the elements in D, the projections between bases. These consist of Clebsch-Gordan coefficients (CGCs); each projection is the product of 4 CGCs, one for each of the two angular momentum couplings and one for each of the two isospin couplings. See the appendix for an explicit derivation of the projections. Each element is weighted by two such projections, one from D and one from D^T. We generate lookup tables of CGCs during the setup phase of our calculations.

We could iterate over the coupled isospins as we do for angular momentum, but the isospin space is always the same size and always small; as discussed in section 4.2, there are only five elements in isospin space. To speed up execution, we unroll the isospin for-loops into into simple

sequential instructions. Inside the angular momentum for-loops, we use simple if-statements to determine the correct CGCs, then add all five elements, weighted accordingly, on one line.

The decoupling algorithm, then, is several precomputations followed by a set of three nested for-loops over coupled angular momentum values. See Algorithm 2 for pseudocode of the entire decoupling function.

Algorithm 2 Decoupling algorithm for full nuclear calculations

```
procedure DECOUPLE(rowState, columnState, potential)
    [CGC_0, CGC_1, CGC_2, CGC_3, CGC_4] ← ISOSPINCGCS(rowState, columnState)
    index ← INDEXINTOPOTENTIAL(rowState, columnState)
    element ← 0
    [J_abmin, J_abmax] ← OUTERROWBOUNDS(rowState)
    [J_a'b'min, J_a'b'max] ← OUTERCOLUMNBOUNDS(columnState)
    for i ← J_abmin to J_abmax do
        for j ← J_a'b'min to J_a'b'max do
            [J_abcmin, J_abcmax] ← INNERBOUNDS(i, j, rowState, columnState)
            for k ← J_abmin to J_abmax do
                sum ← 0
                sum ← sum + CGC_0 potential[index]
                sum ← sum + CGC_1 potential[index + 1]
                sum ← sum + CGC_2 potential[index + 2]
                sum ← sum + CGC_3 potential[index + 3]
                sum ← sum + CGC_4 potential[index + 4]
                index ← index + 5
                sum ← sum · ANGULARMOMENTUMCGCS(i, j, k, rowState, columnState)
                element ← element + sum
            end for
        end for
    end for
    return element
end procedure
```

4.4 Decoupling on the GPU

4.4.1 Level of Parallelism

In this section we present an expanded version of the discussion in Ref. (52). We considered applying GPU acceleration to our decoupling routine at two levels. The simpler, coarse-grained option is to use one GPU thread for each m-scheme element requested by MFDn. The more

complicated fine-grained option is to use multiple GPU threads to calculate the same m-scheme element, parallelizing some or all of the for-loops shown in Algorithm 2.

One disadvantage of the simpler option is the low granularity of parallelization. As discussed in section 4.1, GPUs operate better when there are no branching conditionals or for-loops with different bounds. With the simpler, coarse-grained GPU acceleration approach, we risk allowing GPU threads to sit idle while other threads in the same warp complete longer for-loops. The fine-grained, multiple-threads-per-element scheme offers a possible solution to this issue by dividing up the for-loops into separate chunks. Note that we would have to parallelize all three loops in Algorithm 2 to achieve full benefit of this granularity, as the for-loops are non-rectangular; that is, the size of the inner loop depends on the iteration number of the outer loops.

The coarse-grained option is straightforward to implement, requiring no real changes to Algorithm 2; we move the decoupling routine to the GPU kernel and invoke it on groups of m-scheme element requests from MFDn. The fine-grained method requires a substantially more complicated development effort. In particular, we must deal with the non-rectangular nature of the for-loops. We need to map the GPU thread ID onto a segment of the for-loop space. See section 4.1 for an explanation of why this indexing is necessary for GPU calculations.

Since we cannot easily calculate the number of loop iterations it takes to reach a particular iteration state in the three for-loops, we cannot easily perform this mapping. As an analogy, it is easy to map a single index onto a square grid, but it is much more challenging to map a single index onto a grid that is square on three sides but whose fourth side is determined by a function. Furthermore, we would still expect to calculate many m-scheme elements per kernel invocation with the coarse-grained method. The outer for-loops are of a nonuniform size that depends on the requested m-scheme elements, which are only known when the kernel is invoked; even if we can efficiently calculate the number of loop iterations to reach a particular for-loop state, each GPU thread must perform such a calculation over all lower-index GPU threads to determine its target segment of work.

These difficulties, on top of the development and time-efficiency costs of accumulating coupled-JT element contributions across multiple GPU threads, lead us to adopt the simpler,

coarse-grained approach for our work. We process a number of m-scheme matrix element requests with a single kernel invocation, calculating each m-scheme matrix element with its own GPU thread. Our decoupling code and MFDn both store and reference SPSs with linear indices, so the input to the decoupling code is a block of sets of six SPS indices, flattened into a linear array.

4.4.2 Indexing the Input Interaction

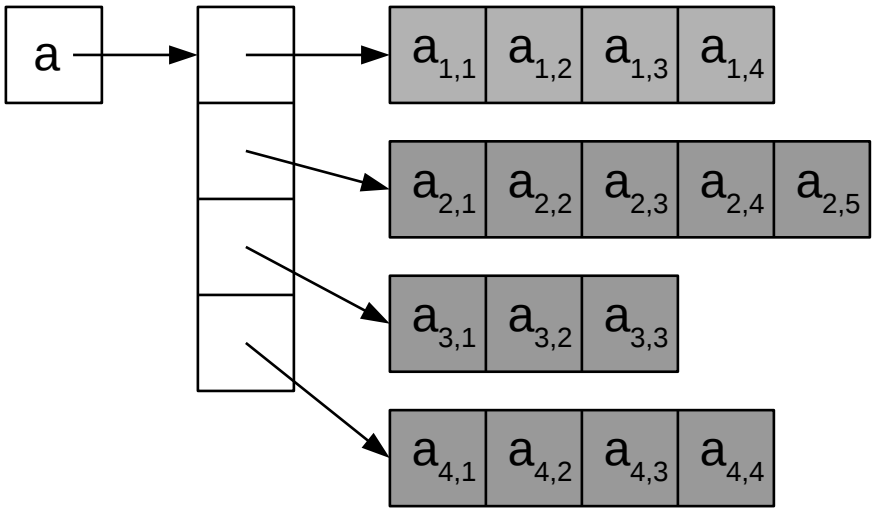

Figure 4.5 A two-dimensional jagged array a stored as a tree of pointers in the C language. We represent pointers as white squares and stored values as gray squares. Values are stored in linear blocks; for each linear block of memory, there is a chain of pointers from the root, and we can locate values even though their memory locations are not at predictable offsets from the memory location of the root pointer. We can therefore store these linear blocks at arbitrary locations and allocate them at arbitrary size, allowing us to store jagged arrays like a.

The non-rectangular nlj-block indexing array presented an issue. The C language supports several methods for storing and accessing multidimensional arrays. A pointer is a reference to a location in memory; arrays in C are stored as pointers to the block of memory that contains their data. If the array is rectangular and all of its dimensions are known when the first (and in this case only) allocation is made, we can initialize the array all at once in a single block of memory, using simple arithmetic to convert between the multidimensional array indices and the single index into the block.

The other method is more general, and relies on a tree of pointers. For a three-dimensional array, for example, the root array is a list of pointers; each of those points to another list of pointers, and each of the pointers in those lists points to a block of actual data, as illustrated in Fig. 4.5. This method is very general; we can allocate earlier dimensions of an array without knowing the bounds of later dimensions, and we can allocate jagged arrays. Reading from it requires an extra memory access for each array dimension beyond one, though, and instead of one memory allocation it requires a number of allocations asymptotically proportional, in the limit of large dimensions, to the total number of elements divided by the size of the last dimension.

Because the nlj-block indexing array was highly jagged, we were forced to rely on this tree-of-pointers method. In the CPU version of the decoupling code, the additional allocations and memory accesses were an acceptable overhead. In the GPU version, we generated the pointer structure on the CPU, then copied it over to the GPU; since it was stored in a fragmented fashion on the CPU, this resulted in an exhorbitant number of small allocations and memory copies on the GPU. GPUs tend to have a higher latency, as discussed in section 4.1, so this produced an unacceptable overhead.

For our first attempt to address this issue, we kept the nlj-block indexing array on the CPU only, and passed block indices to the GPU kernel invocation along with matrix element requests. Unfortunately, this attempt did not produce a favorable speedup.

In our second attempt, we generated the entire index array in a contiguous block on the CPU, producing a structure wherein the pointers were correct relative to each other. We then applied a constant offset to all the pointers in the index array so that their absolute coordinates would be correct for a contiguous block allocated on the GPU. The entire pointer structure could then be copied into that block with one copy, allowing the index structure to be created on the GPU with a single GPU allocate and GPU copy. This improvement reduced the setup time of the nlj-block indexing array to a negligible level, resolving the indexing difficulty. After this improvement, the GPU-accelerated portion of our code executed substantially faster than the original version that ran only on the CPU; see section 4.6 for a quantitative discussion of overall speedup.

Our code relies on several other multidimensional arrays to store the precomputed CGC tables, map SPS indices to SPS quantum numbers, and provide several additional internally useful indexing systems. We generated these on the CPU and copied them to the GPU at the start of our calculations; since they are all rectangular arrays, we were able to prepare them on the GPU with one allocation and copy apiece, and did not face jagged array difficulties there.

Before integrating the GPU-accelerated decoupling code into MFDn as described below in section 4.5, we ran a standalone version to test performance and act as an intermediate step. We compared our standalone GPU code to a multithreaded CPU implementation running on eight cores (51). For our first test case, we used matrix elements corresponding to SPSs from the beginning of the SPS basis, with low SPS indices and low angular momentum. The decoupling loops, which iterate over angular momentum, were therefore rather short, and there was not that much computation to offload to the GPU. We saw a 4x GPU speedup from this less demanding case. Our second test case used matrix elements corresponding to SPSs from further into the SPS basis, with higher angular momenta and longer decoupling loops; this produced a much more computationally intensive test case, and we measured a 10x speedup from our use of GPUs. We have not performed a detailed analysis of which case is a more accurate reflection of our intended usage in MFDn, but we expect MFDn to request m-scheme elements from all parts of the basis.

4.5 MFDn Integration

As discussed in section 3.4, MFDn uses a recursive loop to calculate elements from the full many-body Hamiltonian, occasionally requesting 3-body elements. Many-body elements require differing numbers of 3-body elements, with the number depending on how different the row-state is from the column-state. In the CPU-only version of MFDn, the decoupling code calculates 3-body elements one-by-one, as they are requested. The GPU version of the decoupling code requires a large block of simultaneous requests to be efficient, so the sequential requests in the CPU code are not ideal. To bridge the gap between MFDn and the GPU decoupling code, we use buffers to store lists of 3-body element requests so large chunks of requests can be sent to the GPU at once.

Each OpenMP thread has its own buffer allocated to store requests. On receiving a request, the CPU part of the decoupling subroutine stores the request in the buffer and returns 0. In the CPU-only version, the returned value is added directly to the many-body element under calculation; we must thus also store which many-body element the request pertains to so that it can be added to the correct many-body element when the calculation finishes on the GPU. Furthermore, the 3-body element is added with a specific phase, which must also be stored with the request. Larger buffers are more efficient because the overhead associated with a single kernel invocation and set of memory copies is split over more elements. Tests with the standalone decoupling code running without MFDn indicate diminishing returns above approximately 20,000 elements (51) on the supercomputer Dirac at the National Energy Research Scientific Computing Center. We use buffers of approximately this size here.

Each OpenMP thread starts in accumulating mode, passing element requests to its buffer until it is full. Then, the thread sends the buffer to the GPU and switches into non-accumulating mode. In this mode, the decoupling code runs as in the CPU-only version of MFDn, calculating 3-body elements on the CPU at request and returning them; this allows the CPU to continue work while the GPU, which is shared among many OpenMP threads, is busy. The thread checks periodically for a completed chunk from the GPU. When it receives the chunk, it iterates through the returned three-body elements in the chunk, multiplies them by the stored phases, and adds them to the array of many-body elements at the stored locations. It then switches back into accumulating mode, and the cycle begins again. At the end of the many-body matrix construction phase, all the 3-body contributions have been added in, either from the GPU calculations or directly from the CPU decoupling code.

As discussed in section 3.2, MFDn runs on two MPI processes per Titan node, with eight OpenMP threads per process. We allow GPU requests from both MPI processes so all 16 OpenMP threads can access the decoupling code. By default, only one MPI process can access the GPU on a Titan node; we enable the `CRAY_CUDA_PROXY` option during job submission, which creates a proxy server to manage GPU access from all MPI processes on a node (53). We tried instead running with only one MPI process per node; this splits MPI process over two NUMA domains, incurring a performance penalty as discussed in section 3.2. The resulting

performance penalty is severe enough to deter us from this option. A third option would be to enable GPU acceleration on only one node, running the other node as CPU-only. This presents difficulties with load-balancing, offsetting the nearly perfect loadbalancing that is a design feature of MFDn, so we elect to use the first option here.

4.6 Timing Results

As discussed in section 1.2, we present timing results from the DOE supercomputer Titan at Oak Ridge National Laboratory. We have performed the timing results presented here for nuclear systems, not neutron drops, though we expect speedups for both types of systems. We use the number of nonzero matrix elements in the full many-body Hamiltonian as a measure of problem size, and test at a variety of problem sizes. The number of nonzeroes is determined by the choice of nucleus and $N_{\max}$ truncation parameter, so it is difficult to generate a smooth spectrum of results. Different problem sizes can require vastly different numbers of cores to store the many-body matrix, so we do not test all problem sizes on the same number of nodes; for each problem size, we allow the many-body Hamiltonian to take up half of the total memory, and choose the smallest number of nodes that satisfies that requirement. This is a good approximation to normal use for production runs; the remainder of the memory is used for storing the input interaction file, the Lanczos vectors, Titan's operating system, and other things of that nature. We can allocate more than half of available memory for storing the many-body matrix, but as we increase memory usage beyond half, we run an increasing risk of out-of-memory errors. We report results here from version 14 of MFDn; the GPU-accelerated code uses the "beta 4" release of version 14, and the CPU code uses "beta 6", a slightly later release. Beta 6 incorporates several improvements and optimizations, and was the most advanced production version available at the time. As that version was the most optimized CPU version available, we felt it offered the fairest comparison to the optimizations we performed on the GPU code.

We present timing results as published in Ref. (52). Our primary timing results are summarized in Fig. 4.6. We see a speedup of 2.2x – 2.7x in the matrix construction stage. There is no immediately apparent pattern in the dependence of speedup on problem size; the choices

of nuclei and truncation parameters required to generate the spectrum of problem sizes are somewhat haphazard, so it is possible that any problem size dependence has become entangled with dependence on those parameters. The range of speedups appears to be roughly constant over problem size, suggesting that our GPU acceleration scales well for the problem sizes investigated.

Matrix construction speedup is easy to quantify, but the speedup over the entire run is a more ambiguous quantity. The time taken in the diagonalization stage of MFDn depends on how many eigenvalues are required and their requested accuracy. The speedup over the entire run, which depends on the relative times of the matrix construction and diagonalization stages, therefore depends on these parameters also. For the representative parameter choices used in the matrix construction speedup calculations discussed above in this section, the overall speedup is in the 1.2x – 1.4x range.

We have not yet done detailed timing comparisons for neutron drop calculations on the GPU. The isospin space for nuclei is five times larger than the isospin space for neutron drops, as can be seen in Fig. 4.3, noting that neutron-only 3-body MBSs must have total isospin $T = 3/2$. Inside the nested for-loops in Algorithm 2, neutron drop calculations have one contribution instead of five. This means that neutron drop decoupling calculations are less computationally intensive than decoupling calculations for nuclei. While we would expect the speedup of the decoupling process itself for neutron drops to be comparable to the results reported above, the decoupling for the neutron drops contributes less to the overall time, so we would expect a correspondingly smaller overall speedup.

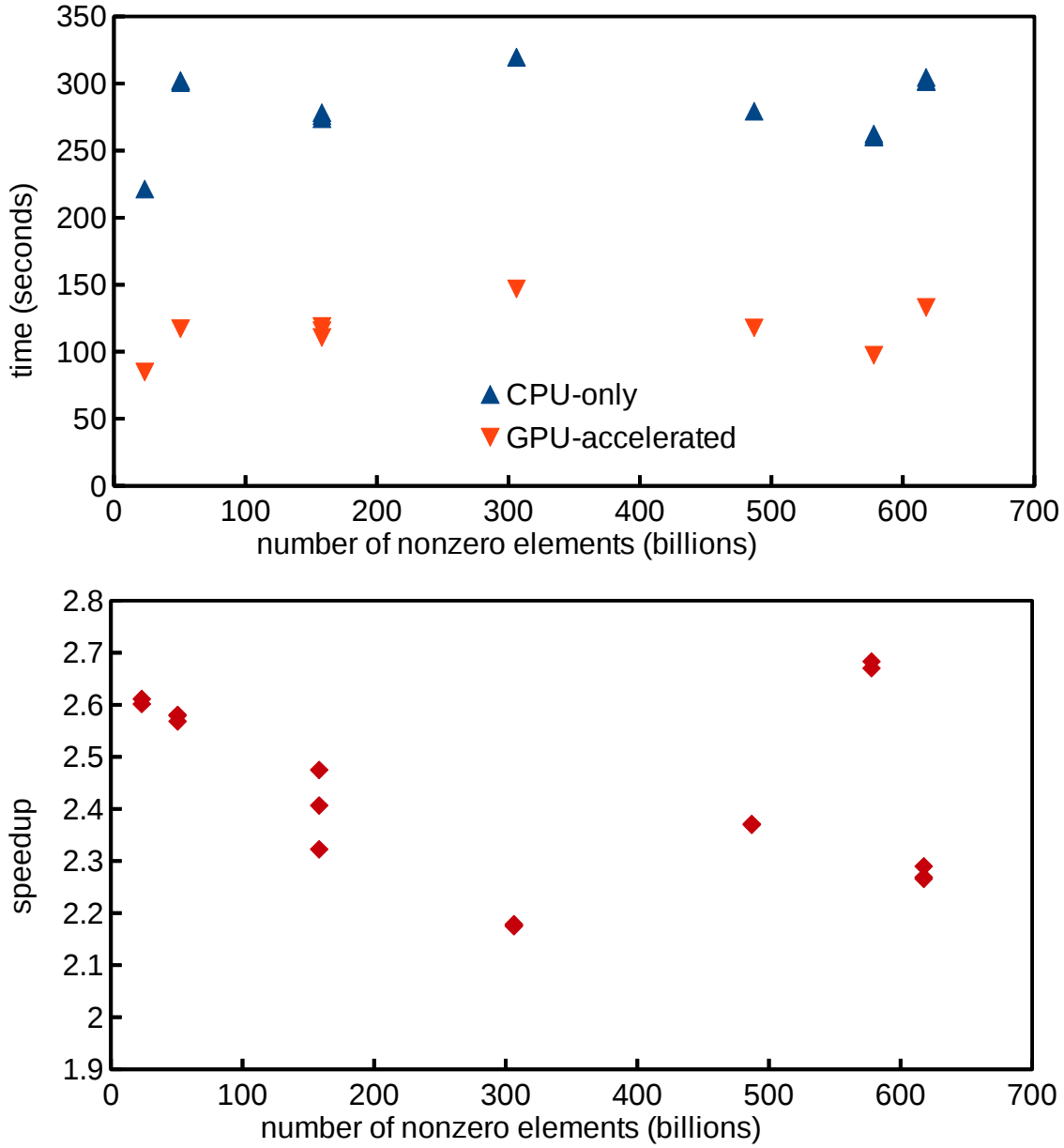

Figure 4.6 (Color online.) Matrix construction phase times (upper panel) and GPU-accelerated speedup (lower panel) for nuclei calculations using CPU-only and GPU-accelerated versions of MFDn Figures adapted from Ref. (52).

CHAPTER 5. NEUTRON DROP RESULTS

5.1 Overview

We present our physics results for neutron drops in this chapter. Section 5.2 summarizes some of the results of the study in Ref. (5). These results apply to a smaller range of neutron drop systems and external harmonic potentials, and serve to illustrate important aspects of the SRG transformation discussed in section 2.4 and the effects of the chiral 3N force. In the sections following, we then present results for our full range of neutron drop calculations, with systems from $N = 2$ to 40 neutrons in 10 MeV and 20 MeV external harmonic potentials.

For comparison in both sections, we show our results with the AV8′ plus 3N interactions (UIX and IL7) QMC results from Refs. (3; 4). (Ref. (4) provides multiple states with different J for odd neutron drops; in all of our figures with AV8′, AV8′+UIX or AV8′+IL7 we use the states with lowest total energy for comparison with our results.) We present AFDMC results for energies, energy differences, and level splittings; for internal energies and radii, we present a mix of AFDMC and GFMC results, following the choices of Ref. (4). In addition, we present results from the JISP16 (30; 31), a nonlocal NN potential without 3N interactions; the JISP16 results presented here have appeared previously in Ref. (4).

Note that Ref. (5) uses the term "No-Core Shell Model" (NCSM) for the same formalism that we refer to as NCFC in this work and describe in chapter 2. For this current work, we follow the usage of Ref.s (10; 8), which use the name NCSM for an earlier variant (8) of the NCFC formalism that makes use of a similarity transformation in addition to the methods described in chapter 2. Neither this work nor the study in Ref. (5) uses the similarity transformation discussed in Ref. (8).

5.2 3N Forces and SRG Behavior

Table 5.1 Comparison of total ground-state energies in units of MeV for $N = 2$ through 18 neutrons in a 10 MeV HO trap. We use the SRG-evolved NN+3N-full Hamiltonian, which includes the initial chiral 3N interaction at SRG evolution scales $\alpha = 0.04$ fm^4 and 0.08 fm^4. We also present results for the NN+3N induced Hamiltonian without initial chiral 3N at SRG evolution scale $\alpha = 0.08$ fm^4. Uncertainties, as explained in the text, are quoted in parenthesis for the last significant figure. We present unextrapolated NCFC results at $N_{\max} = (14, 12, 10, 8, 6)$ for $N = (2, 3-4, 5-9, 10-16, 17-18)$. These results have been reproduced from Ref. (5).

		3N-full		3N-ind.
N	J^π	$\alpha = 0.04$ fm^4	$\alpha = 0.08$ fm^4	$\alpha = 0.08$ fm^4
2	0^+	23.88(1)	23.897(1)	23.897(1)
3	$3/2^-$	45.51(2)	45.534(4)	45.532(4)
4	0^+	62.17(4)	62.207(8)	62.133(7)
5	$3/2^-$	82.65(7)	82.67(3)	82.63(3)
6	0^+	98.6(1)	98.62(4)	98.46(4)
7	$1/2^-$	118.2(1)	118.24(6)	117.97(5)
8	0^+	135.4(2)	135.34(7)	134.42(7)
9	$5/2^+$	162.8(2)	162.8(1)	161.84(9)
10	0^+	187.1(6)	186.8(4)	185.6(3)
11	$3/2^+$	213.6(7)	213.3(4)	212.2(4)
12	0^+	237.1(8)	236.7(5)	235.3(4)
13	$5/2^+$	263.0(9)	262.6(6)	261.2(5)
14	0^+	286(1)	285.3(7)	283.7(6)
15	$1/2^+$	311(1)	310.1(7)	308.6(6)
16	0^+	334(1)	333.1(8)	331.2(7)
17	$3/2^+$	361(2)	360(2)	357(2)
18	0^+	386(3)	385(2)	381(2)

In this section, we present results from an initial study (5) to justify our use of the SRG transformation and comment on the strength of chiral 3N effects. As discussed in section 2.4, we use NN+3N chiral potentials softened with an SRG transformation. The SRG transformation induces higher-body terms, which we ignore in our present application. For a select subset of our full results, we have performed runs at SRG flow parameter $\alpha = 0.04$ fm^4 in addition to our full results at $\alpha = 0.08$ fm^4; this gives us a check on whether we are justified in neglecting the induced higher-body terms. In this smaller study, we have also performed calculations with no explicit 3N forces. That is, "3N-induced" calculations where our potential has no 3N contribution before the SRG transformation, but where we retain the 3N contribution induced from lower-body terms by the SRG transformation. This offers insight into the importance of chiral 3N forces in the $T = 3/2$ channel.

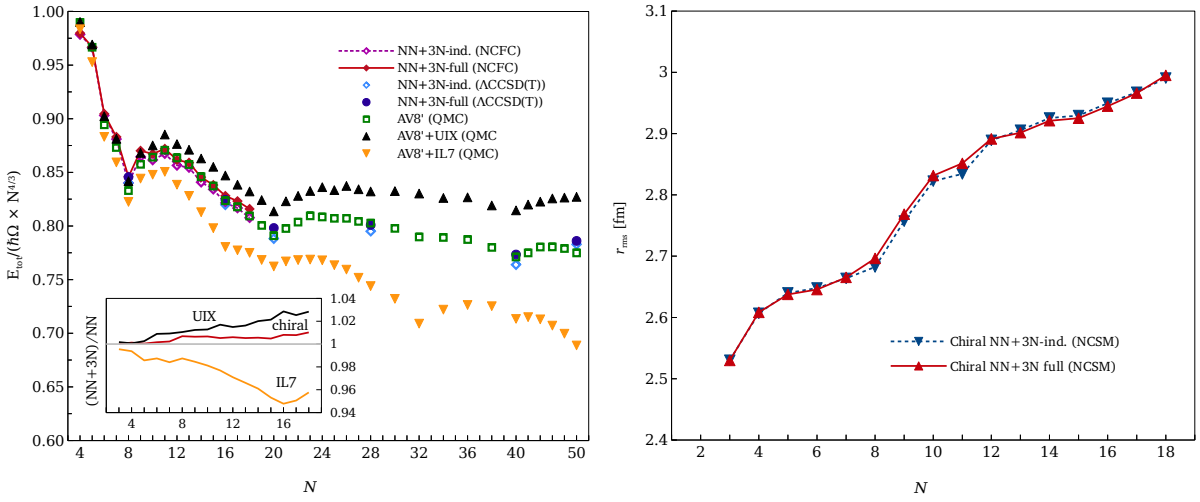

Figure 5.1 (Color online.) Scaled total energies (left panel) and unscaled radii (right) of N-neutron systems in a 10 MeV harmonic trap for different Hamiltonians as a function of N. The results for the chiral interactions are obtained in the NCFC formalism with the largest accessible $N_{\max}$ and for closed subshells in coupled cluster theory. The results labeled AV8′, AV8′+UIX and AV8′+IL7 are adopted from Ref. (4), and the chiral results and figure are reproduced from Ref. (5).

We present numerical results for the ground state energies in Table 5.1, adapted from Ref. (5). As in Ref. (5), for this initial study we eschew more complicated extrapolation methods like the ones described in sections 5.3 and 5.5. We present basis parameter $\hbar\omega = 16$ MeV results at the highest $N_{\max}$ value available, taking the uncertainty to be the difference between the highest and second-highest-$N_{\max}$ values. We present results in a 10 MeV trap only, investigating neutron drop systems from $N = 2$ to 20.

We can immediately see that changing the SRG flow parameter from 0.04 fm^4 to 0.08 fm^4 has very little effect; the highest percentage difference between the $\alpha = 0.04$ fm^4 and $\alpha = 0.08$ fm^4 results is 0.29% at $N = 15$. In light of these results, we are confident in proceeding with the $\alpha = 0.08$ fm^4 SRG transformation for our larger body of results discussed in the remaining sections of this chapter; this choice provides results closer to convergence with the available computational resources.

We note also that the chiral 3N force has a surprisingly weak effect. This is more evident in the left panel of Fig. 5.1, where we plot the chiral ground state energies with and without 3N potentials. The SRG flow parameter dependence is not visible at that scale, so we plot only

Table 5.2 Total ground-state energies $E_{\rm tot}$ and correlation energies ΔE in units of MeV at closed subshells obtained with the chiral interactions in coupled-cluster theory at the ΛCCSD(T) level for a 10 MeV HO trap, reproduced for comparison from Ref. (5). We use the SRG-evolved NN+3N-full and NN+3N-induced Hamiltonian at SRG evolution scales $\alpha = 0.04$ fm^4 and 0.08 fm^4 as in Table 5.3. Uncertainties are quoted in parenthesis for the last significant figure; see Ref. (5) for further details.

	ΛCCSD(T)							
	3N-full				3N-ind.			
	$\alpha = 0.04$ fm^4		$\alpha = 0.08$ fm^4		$\alpha = 0.04$ fm^4		$\alpha = 0.08$ fm^4	
N	$E_{\rm tot}$	ΔE	$E_{\rm tot}$	ΔE	$E_{\rm tot}$	ΔE	$E_{\rm tot}$	ΔE
8	135.1(1)	-3.7	135.3(1)	-3.0	134.2(1)	-3.6	134.4(1)	-2.9
16	332.0(6)	-11.1	332.4(5)	-8.5	329.9(7)	-10.9	330.5(5)	-8.6
20	432.8(5)	-11.1	433.3(3)	-7.7	427.1(4)	-9.9	427.9(2)	-6.8
28	681(1)	-19.6	681(1)	-14.0	674(1)	-17.8	676(1)	-13.0
40	1058(2)	-25.9	1058(1)	-16.4	1043(2)	-20.9	1045(1)	-12.9
50	1449(3)	-36.6	1448(2)	-23.3	1438(5)	-29.4	1442(4)	-18.3

the $\alpha = 0.08$ fm^4 results. The UIX and IL7 3N forces have a much stronger effect than the chiral forces. We illustrate this effect in the left-panel inset of Fig. 5.1, which shows the chiral and phenomenological 3N force energies normalized by their corresponding NN-only energies.

Such a wide range of 3N effects in the total energies - decreasing energies for AV8′+IL7, increasing energies for AV8′+UIX, and a lack of sensitivity for chiral 3N forces - indicates that there is a significant opportunity for improvement in the $T = 3/2$ channel of the nuclear potentials examined here. We expect neutron-rich nuclei to be an excellent tool for exploring this variance in a more realistic and experimental context.

The striking chiral insensitivity to 3N forces extends also to the root-mean-square (rms) radii, plotted in the right panel of Fig. 5.1. Since the 10 MeV external harmonic potential that we impose for these results is a scaled version of the $r^2_{\rm rms}$ operator, rms radius serves as a measure of how much of the total energy is from the external harmonic potential and how much is from kinetic energy and inter-nucleon potentials. The insensitivity of the radii to chiral 3N forces indicates that not only total energy, but also this energy balance is rather insensitive to chiral 3N forces. For the full set of neutron drop systems and external trap strengths presented in the remaining sections of this chapter, we use chiral forces with full 3N contributions only.

In Table 5.2, reproduced from Ref. (5), we present additional calculations performed in the coupled cluster (CC) formalism (54; 55; 56; 57; 58; 59) with same the chiral Hamiltonians used in our NCFC results. The CC formalism is capable of providing results for heavier systems,

especially at subshell closures; for more details concerning the CC results reproduced here, see Ref. (5). These CC results provide both a check of our NCFC methodology and an indication that the 3N insensitivity of our chiral Hamiltonians persists into heavier systems.

5.3 Ground State Energies

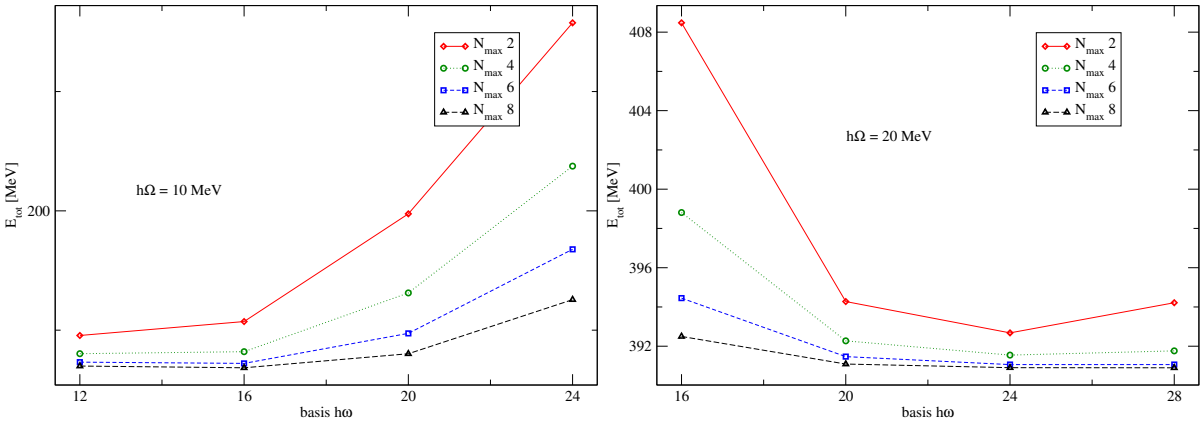

Figure 5.2 (Color online.) Convergence pattern in $N_{\max}$ and basis $\hbar\omega$ of chiral NN+3N NCFC ground state total energy results for a 10-neutron system in 10 MeV and 20 MeV harmonic traps.

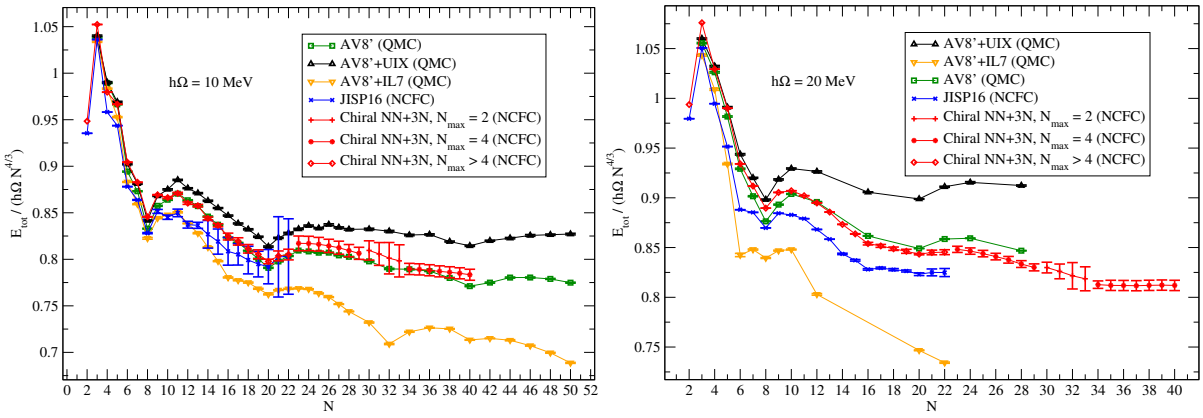

Figure 5.3 (Color online.) Total energy (scaled) of N-neutron systems in 10 MeV and 20 MeV harmonic traps for different Hamiltonians as a function of N. The results for the chiral interactions are obtained in the NCFC formalism with the largest accessible $N_{\max}$ (cf. Table 5.3) and for closed subshells in ΛCCSD(T) (cf. Table 5.2). The results labeled AV8′, AV8′+UIX and AV8′+IL7 are adopted from Ref. (4).

We begin the presentation of our full range of results with our ground state energy results. For each neutron system we perform calculations at a range of basis $\hbar\omega$ parameters and in-

Table 5.3 Comparison of total ground-state energies in units of MeV for $N = 2$ through 40 neutrons in 10 MeV and 20 MeV HO traps. We use the SRG-evolved NN+3N-full Hamiltonian, which includes the initial chiral 3N interaction at SRG evolution scale $\alpha = 0.08$ fm^4, corresponding to momentum scale $\lambda_{\rm SRG} = 1.88$ fm^{-1}. Extrapolation uncertainties, as explained in the text, are quoted in parenthesis for the last significant figure; $N_{\rm max} = 2$ and 4 results are upper bounds only. The results with no errors listed are well-converged, with an extrapolation error below 1 keV.

		10 MeV trap		20 MeV trap	
N	$N_{\rm max}$	J^π	$E_{\rm tot}$	J^π	$E_{\rm tot}$
2	14	0^+	23.897	0^+	50.083
3	12	$3/2^-$	45.532(3)	$3/2^-$	93.107
4	12	0^+	62.200(7)	0^+	130.7276(4)
5	10	$3/2^-$	82.65(1)	$3/2^-$	169.2916(4)
6	10	0^+	98.60(2)	0^+	203.689(1)
7	10	$1/2^-$	118.21(3)	$1/2^-$	244.197(2)
8	10	0^+	135.30(4)	0^+	284.672(5)
9	10	$5/2^+$	162.74(6)	$5/2^+$	338.997(7)
10	8	0^+	186.6(2)	2^+	390.85(5)
11	8	$3/2^+$	213.0(3)	$3/2^+$	441.39(7)
12	8	0^+	236.4(4)	2^+	491.67(9)
13	8	$5/2^+$	262.1(4)	$5/2^+$	541.5(1)
14	8	0^+	284.8(5)	0^+	589.3(1)
15	8	$1/2^+$	309.5(6)	$1/2^+$	639.0(2)
16	6	0^+	332(1)	0^+	689(1)
17	6	$3/2^+$	358(2)	$3/2^+$	745(1)
18	6	0^+	383(2)	2^+	801(2)
19	6	$3/2^+$	408(2)	$3/2^+$	858(2)
20	8	0^+	433(1)	0^+	915.7(7)
21	6	$7/2^-$	466(3)	$7/2^-$	979(3)
22	6	0^+	496(4)	2^+	1042(3)
23	4	$5/2^-$	534(5)	$5/2^-$	1110(4)
24	4	2^+	566(5)	4^+	1172(4)
25	4	$5/2^-$	597(5)	$5/2^-$	1234(5)
26	4	2^+	627(5)	2^+	1295(5)
27	4	$1/2^-$	658(6)	$7/2^-$	1357(5)
28	4	0^+	688(6)	0^+	1417(6)
29	4	$3/2^-$	718(6)	$3/2^-$	1479(6)
30	2	2^+	755(1)	2^+	1547(1)
31	2	$3/2^-$	785(12)	$3/2^-$	1609(14)
32	2	0^+	814(17)	0^+	1669(27)
33	2	$1/2^-$	845(18)	$1/2^-$	1733(26)
34	4	0^+	870(7)	0^+	1790(8)
35	4	$5/2^-$	903(7)	$5/2^-$	1859(12)
36	4	2^+	937(7)	2^+	1930(12)
37	4	$3/2^-$	970(7)	$3/2^-$	2001(13)
38	4	2^+	1004(8)	2^+	2075(13)
39	4	$5/2^-$	1038(8)	$5/2^-$	2148(14)
40	4	0^+	1072(8)	0^+	2221(14)

creasing $N_{\max}$ truncation parameters. As guaranteed by the variational principle, the results converge uniformly from above for all values of $\hbar\omega$ as $N_{\max}$ increases; we accept our results as converged when they are independent of $\hbar\omega$ and $N_{\max}$.

Computational limits often prevent us from achieving a large enough model space to directly converge our results, so we adopt an exponential model of convergence(10; 60) and extrapolate to the infinite model space. We assume that the ground state energies depend on $N_{\max}$ according to

$$E_{\rm tot} = Ae^{-BN_{\max}} + C \tag{5.1}$$

and fit A, B, and C to the three highest-$N_{\max}$ results available. The extrapolated energy is then given by C. This seems to be a reasonable functional form for the extrapolation, as discussed in Refs. (10; 61; 62). For the results presented here, we take the difference between the extrapolation and the highest-$N_{\max}$ result to be our extrapolation error. For cases where we have only two $N_{\max}$ values, we present an upper bound only. For the ground state energy results cited here, we perform extrapolations at all calculated values of basis $\hbar\omega$ and choose the one with the minimal error. We use this extrapolation method where we have results through $N_{\max} = 6$. For the lower-$N_{\max}$ systems beyond $N = 22$, we cite the highest-$N_{\max}$ value at the basis $\hbar\omega$ where it is lowest as an upper bound, using the difference between the highest- and second-highest-$N_{\max}$ values as the extrapolation error.

We present our results for total ground state energy in the 10 and 20 MeV traps in Fig. 5.3. Following the practice of Refs. (2; 3; 4), we scale our total energy plot by the Thomas-Fermi N-dependence ($N^{4/3}$) and by the HO well strength. $N^{4/3}$ is an estimate of the number of HO quanta required, per the Pauli exclusion principle, to support N identical, noninteracting particles in a harmonic potential; to within constant factors, n full HO energy shells contain n^3 identical particles and n^4 HO quanta. In this sense, we are scaling by an approximation of the non-interacting limit of a pure HO many-body Hamiltonian. For comparison, we show our results with the AV8′ plus UIX or IL7 3N force results as discussed in section 5.2. The left panel of Fig. 5.3 is an updated version of the left panel of Fig. 5.1; it features extrapolated NCFC results where feasible and error estimates for all NCFC results, and it extends the range of NCFC results through $N = 40$.

We find a pronounced dip due to the expected shell closure at $N = 8$ in the total energies for the chiral interactions in both the 10 MeV trap and the 20 MeV trap, similar to the dips seen in the other interactions. At $N = 20$, we see good evidence of the shell closure for chiral results in the 10 MeV trap, but only a faint indication in the 20 MeV trap. We note that in the 10 MeV trap, the chiral results are very close to AV8′ and near or between AV8′+UIX and AV8′+IL7 across the entire range of results. Because the chiral 3N forces have only a small effect on the ground state energies (5), we can interpret this as an agreement between the AV8′ and chiral NN forces. We see a similar agreement in the 20 MeV trap, though the AV8′ results are slightly farther from the chiral results, and, as with the chiral results, appear to have a less pronounced shell closure at $N = 20$.

We tabulate our extrapolated ground-state energies (unscaled) in Table 5.3. Comparing these total energy results with the $\alpha = 0.08$ fm^4 3N-full results in Table 5.2 from section 5.2 for the cases that occur in both tables, we observe a good quantitative agreement of CC and NCFC results. The largest fractional difference for a case found in both tables is at $N = 40$, where the NCFC result is an upper bound only and the difference is 1.5%; for all cases where the NCFC results are extrapolated, CC and NCFC results agree to within the precision of the less precise result of the two.

5.4 Energy Differences

The energy differences between neighboring systems in N are shown in Fig. 5.4 for several Hamiltonians. The energy differences act as a low-resolution derivative, allowing us to more readily observe features that are less obvious in the total energies of section 5.3. Pairing effects are easier to see in the energy differences, and energy subshells show up as regions of approximately constant energy difference; the height differences between these regions serve as an indicatation of the strength of the corresponding subshell closures. We can interpret the energy difference between an N-neutron system and a system with $N - 1$ neutrons as the energy cost to remove a neutron from the N-neutron system, and, consequently, as a proxy for the neutron removal energy of corresponding neutron-rich nuclear systems.

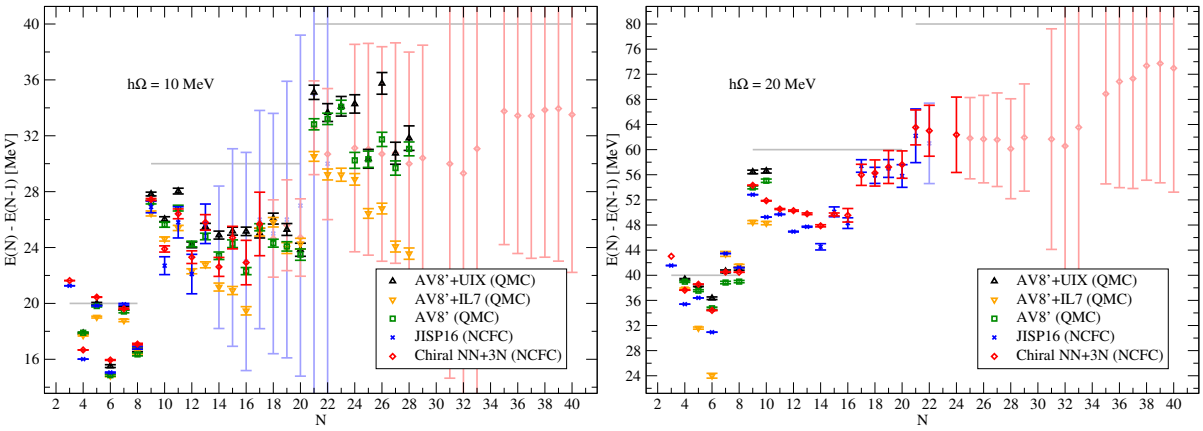

Figure 5.4 (Color online.) Total ground-state energy differences for neutron drops with neighboring numbers of neutrons for various interactions as indicated in the legend. Horizontal gray lines indicate the differences expected from a pure harmonic potential with no interactions. The results for AV8′+3N and JISP16 are taken from Ref. (4). Results with above 10% error are displayed lighter.

We calculate these energy differences from the extrapolated ground state energy results discussed in the previous section. We use a standard error calculation to propagate the error from the energies to the energy differences; to calculate the error δf in a function f of independent variables x_i, we scale the individual errors δx_i by the partial derivatives $\frac{\partial f}{\partial x_i}$ and sum in quadrature:

$$\delta f = \sqrt{\sum_i \left(\delta x_i \frac{\partial f}{\partial x_i}\right)^2} \tag{5.2}$$

Unless otherwise specified, we use this calculation for all error propagation in this work. In the chiral results, we have three different methods for obtaining final energy results: exponential extrapolation and two different qualities of upper bound. We expect systematic differences between these methods, so we discard energy difference results that rely on two types of energy results. As discussed in Section 5.3, the chiral ground state energies above $N = 22$ are upper bounds only, and have large error bars. In addition, the differences are much smaller than the energies themselves, but the difference uncertainties are not suppressed in the same way. Each difference uncertainty also includes contributions from two energy uncertainties. The uncertainties that we see in the energies are therefore substantially magnified in the differences. We see similar effect in the double differences, discussed later in this section, as well.

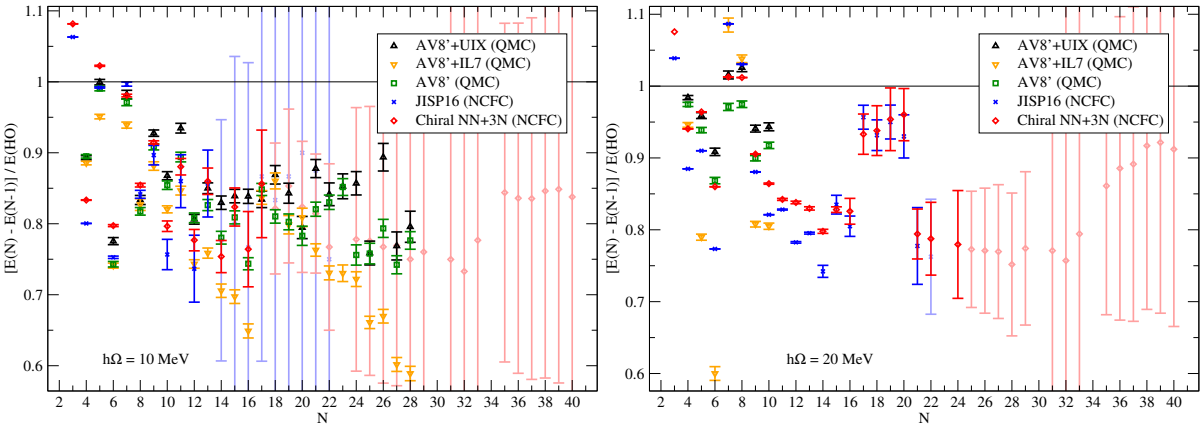

Figure 5.5 (Color online.) Total ground-state energy differences for neutron drops with neighboring numbers of neutrons for various interactions as indicated in the legend, scaled by the values expected from a pure harmonic potential with no interactions. The results for AV8′+3N and JISP16 are taken from Ref. (4). Results with above 10% error are displayed lighter.

In some ranges, the interactions produce the strong odd-even effect conventionally characterized as a "pairing energy" effect. In the 10 MeV trap we see the effects of pairing clearly in the p-shell (3-8 neutrons) and the sd-shell (9-20 neutrons), but in the pf-shell (21-40 neutrons) pairing effects are less prominent. In the 20 MeV trap we see little evidence of pairing in the p-shell, and no evidence of pairing in the higher shells. We see a good agreement between the different interactions in the p-shell in the 10 MeV trap, but in higher shells and in the 20 MeV trap results diverge more.

In the 10 MeV trap, the AV8′+IL7 results show a clear discontinuity at the $N = 16$ subshell closure (with closed orbitals $0s_{1/2}$, $1p_{3/2}$, $1p_{1/2}$, $2d_{5/2}$, and $2s_{1/2}$). We do not have AV8′+IL7 energy difference results beyond $N = 28$, but the even-N AV8′+IL7 total energy results in the 10 MeV trap in Fig. 5.3 suggest a strong subshell closure at $N = 32$ (with closed orbitals $3f_{7/2}$ and $3p_{3/2}$ in addition to the closed s, p and sd energy shells). For both cases, the behavior in the lower subshells of the sd shell is very similar to the behavior in the lower subshells of the pf energy shell: a consistent decrease before the subshell closure. By comparison, the other interactions have flatter energy differences in those lower subshells. In the chiral results, we note what appears to be a sharp discontinuity at the supposed $N = 32$ subshell closure.

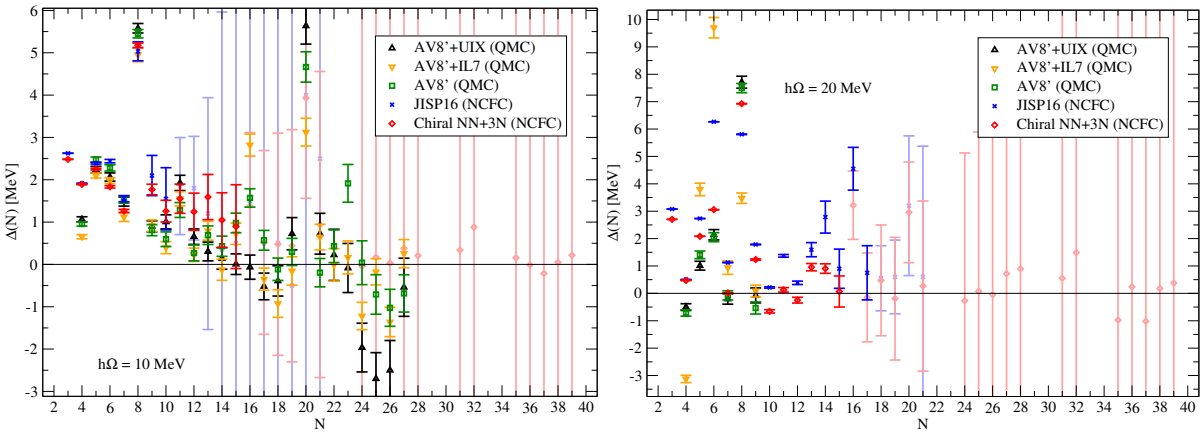

Figure 5.6 (Color online.) Total ground-state energy double differences for various interactions as indicated in the legend. The results for AV8′+3N and JISP16 are taken from Ref. (4). Some results with higher error are displayed lighter.

Traditionally, we expect the $f_{7/2}$ shell to have a strong subshell closure, corresponding to a magic number at $N = 28$; our results suggest that $p_{3/2}$ is strongly closed instead, implying a magic number at $N = 32$. This suggests that ^{52}Ca might be a better candidate doubly magic nucleus than ^{48}Ca, the traditionally identified doubly magic nucleus. We are missing the chiral $N = 34$ point, however, and the errors in that region are large, so we are hesitant to draw firm conclusions.

In the 20 MeV trap, we see evidence of the subshell closures at $N = 6$ for all interactions, and at $N = 16$ in the chiral and JISP16 results as well. The supposed subshell closure at $N = 32$ is also evident in the chiral results, but less clearly so; the pf-shell energy differences rise less abruptly after $N = 32$, and the errors are again large in that region. Overall, we see that the $N = 6$ and 16 shell closures are stronger in the 20 MeV trap than the 10 MeV trap, with the possible exception of AV8′+IL7, for which we do not have sufficient 20 MeV trap results. See section 5.6 for further discussion of this topic in the context of level splittings.

Without interactions, we expect the single differences to be simple multiples of the HO trap energy as indicated by the solid horizontal lines. In Fig. 5.5, we show the single differences scaled by the noninteracting limit, showing more clearly the departure from noninteracting behavior. After the p shell, results in both traps are below the noninteracting limit, as we would expect from the attractive nature of the inter-nucleon strong force at medium range.

For the chiral results in the 20 MeV trap, the higher subshell in the *sd* shell appears as a deviation from the reasonably smooth results on either side; the JISP16 results show a similar, though less pronounced effect.

In some ranges, the interactions produce the strong odd-even staggering effect conventionally characterized as a "pairing energy" effect. When we have an even number of neutrons, we expect each neutron to "pair" with another in a more stable low-energy configuration, reducing the overall energy of the system. In the 10 MeV trap we see the effects of pairing clearly in the *p*-shell and the *sd*-shell, but in the pf-shell pairing effects are less prominent. In the 20 MeV trap we see little evidence of pairing in the *p*-shell, and no evidence of pairing in the higher shells. We see a good agreement between the different interactions in the *p*-shell in the 10 MeV trap, but in higher shells and in the 20 MeV trap results diverge more.

We illustrate the pairing behavior by presenting the double differences $\Delta(N)$, defined as

$$\Delta(N) = (-1)^{(N-1)}(E(N) - \frac{1}{2}(E(N-1) + E(N+1))) \tag{5.3}$$

which are shown in Fig. 5.6. $\Delta(N)$ functions approximately as a second derivative, measuring concavity with the sign reversed for even N. With the expected pairing behavior, we expect even-N systems to have lower total energy than their odd-N neighbors; with the sign reversal, then, positive $\Delta(N)$ indicates expected pairing behavior. The chiral results are largely positive in the 10 MeV trap, with negative results for a few neutron numbers. In the 20 MeV trap, chiral results are less strongly positive, consistent with our observations of the single differences.

In the 10 MeV trap, we see a clear spike in $\Delta(N)$ with all interactions for the shell closure at $N = 8$; we see a similar spike in the 20 MeV trap for some interactions, but JISP16 and AV8′+IL7 have a higher spike nearby at $N = 6$. The $N = 20$ shell closure is clearly visible in the 10 MeV trap for all interactions, and less clearly visible in the 20 MeV trap for the available interactions. By contrast, the $N = 16$ subshell closure is more visible for the chiral results in the 20 MeV trap than in the 10 MeV trap.

Table 5.4 Comparison of rms radii for $N = 2$ though 22 neutrons. Our extrapolation method requires $N_{\max} = 6$ results for good performance; at present, we omit systems where $N_{\max} = 6$ results and higher are unavailable. Estimated numerical uncertainties, as discussed in the text, are included in parentheses.

		10 MeV trap		20 MeV trap	
N	$N_{\max}$	J^π	$r_{\rm rms}$	J^π	$r_{\rm rms}$
2	14	0^+	2.28	0^+	1.68
3	12	$3/2^-$	2.5309(5)	$3/2^-$	1.84
4	12	0^+	2.6092(5)	0^+	1.92
5	10	$3/2^-$	2.638(2)	$3/2^-$	1.93
6	10	0^+	2.646(3)	0^+	1.94
7	10	$1/2^-$	2.667(5)	$1/2^-$	1.98
8	10	0^+	2.699(5)	0^+	2.02
9	10	$5/2^+$	2.772(7)	$5/2^+$	2.07
10	8	0^+	2.842(9)	2^+	2.101(1)
11	8	$3/2^+$	2.86(1)	$3/2^+$	2.125(1)
12	8	0^+	2.91(1)	2^+	2.147(1)
13	8	$5/2^+$	2.92(2)	$5/2^+$	2.162(5)
14	8	0^+	2.94(2)	0^+	2.175(6)
15	8	$1/2^+$	2.95(3)	$1/2^+$	2.189(8)
16	6	0^+	2.97(5)	0^+	2.208(4)
17	6	$3/2^+$	3.00(5)	$3/2^+$	2.232(9)
18	6	0^+	3.03(5)	2^+	2.26(1)
19	6	$3/2^+$	3.05(5)	$3/2^+$	2.28(1)
20	8	0^+	3.07(2)	0^+	2.298(1)
21	6	$7/2^-$	3.10(5)	$7/2^-$	2.32(1)
22	6	0^+	3.13(6)	2^+	2.33(1)

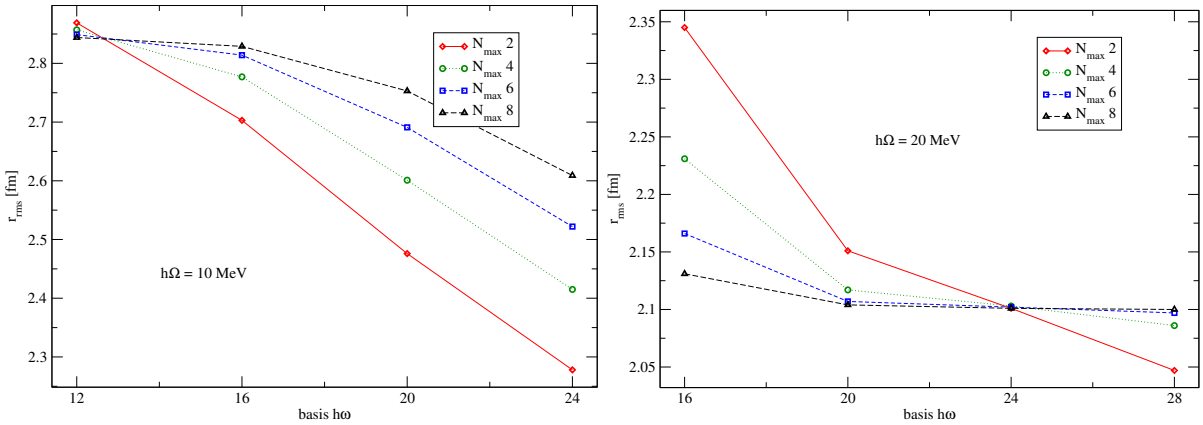

Figure 5.7 (Color online.) Convergence pattern in $N_{\max}$ and basis $\hbar\omega$ of chiral radius results for a 12-neutron system in 10 MeV and 20 MeV harmonic traps.

5.5 Radii and Internal Energies

We move next to the single-particle root-mean-square (rms) radii. As discussed in section 5.2, the external harmonic potential is a scaled version of the $r^2_{\rm rms}$ operator, so the radius serves as an indication of how the total energies are divided between the external harmonic trap and the internal kinetic energy and inter-nucleon potential. In addition, a recent theoretical study (6) investigates the correlation between neutron drop radii and neutron skin thickness in ^{48}Ca and ^{208}Pb; the study shows a strong correlation across many different interactions, using both *ab initio* methods and relativistic and non-relativistic density functional theory. We therefore expect neutron drop radii, when supplemented with calculations of neutron skin thickness in nuclei, to offer an important comparison to experiment through this theoretical link.

As shown in Fig. 5.7, the rms radii converge from both above and below, depending on the value of basis $\hbar\omega$. Simple exponential extrapolations are less effective here than for the energies. Instead, we note that the curves for successive values of $N_{\max}$ tend to cross in the region of basis $\hbar\omega$ where the radii are best-converged - i.e., where the radii convergence pattern is flattest. We therefore take the intersection between the highest- and second-highest $N_{\max}$ radial curves over $\hbar\omega$ to be our approximation to the converged rms radius result, as suggested in Refs. (22; 63) and investigated in Ref. (64; 65). We perform cubic interpolations between successive values of basis $\hbar\omega$ to better capture the basis parameter dependence, and estimate

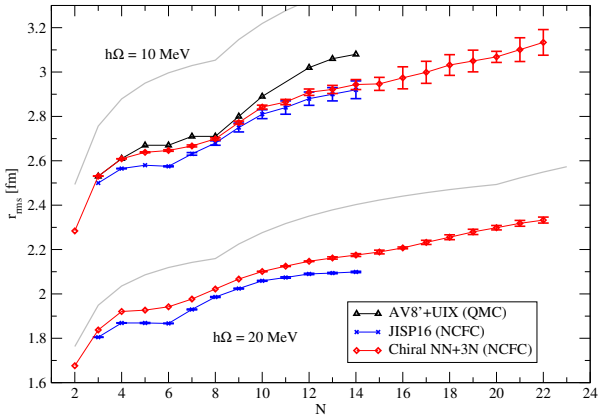

Figure 5.8 (Color online.) Single-particle rms radii for the lowest energy states of $N = 2 - 22$ systems in 10 MeV and 20 MeV HO traps for various interactions as indicated in the legend. Gray lines indicate the radii expected from noninteracting neutrons in a pure harmonic potential. The results for AV8′+UIX and JISP16 are taken from Ref. (4) (note that Fig. 9 of Ref. (4) shows the rms radii of the states with lowest J, which are not always the lowest energy states).

the numerical uncertainty as proportional to the slope of the highest-$N_{\max}$ cubic interpolation at the point of convergence. We apply the cubic interpolation method to systems with results at $N_{\max} = 6$ or higher.

We present the single-particle root-mean-square (rms) radii in Fig. 5.8 for a selection of Hamiltonians. Chiral radii increase consistently with N, showing no odd-even staggering effects for either external trap strength. JISP16 and AV8′+UIX exhibit qualitatively similar monotonic curves, though AV8′+UIX has a slight indication of odd-even staggering in the p-shell. AV8′+UIX generally has a larger radius in the 10 MeV trap; JISP16 matches the chiral results closely in the 10 MeV trap after $N = 8$, but is generally lower over the entire range in the 20 MeV trap.

We see a flatter region around $N = 5$ and another region with reduced slope around $N = 14$ for all interactions at both trap strengths. We can explain this broad structure through comparison with results in the non-interacting limit, plotted in gray on Fig. 5.8. The first neutron in a new shell has a substantially higher $\langle r^2_{\rm rms}\rangle$, which increases the average $\langle r^2_{\rm rms}\rangle$ quickly. As more neutrons are added to the new shell, the average moves closer to $\langle r^2_{\rm rms}\rangle$ for the neutrons residing only in that shell, and so increases less rapidly with N.

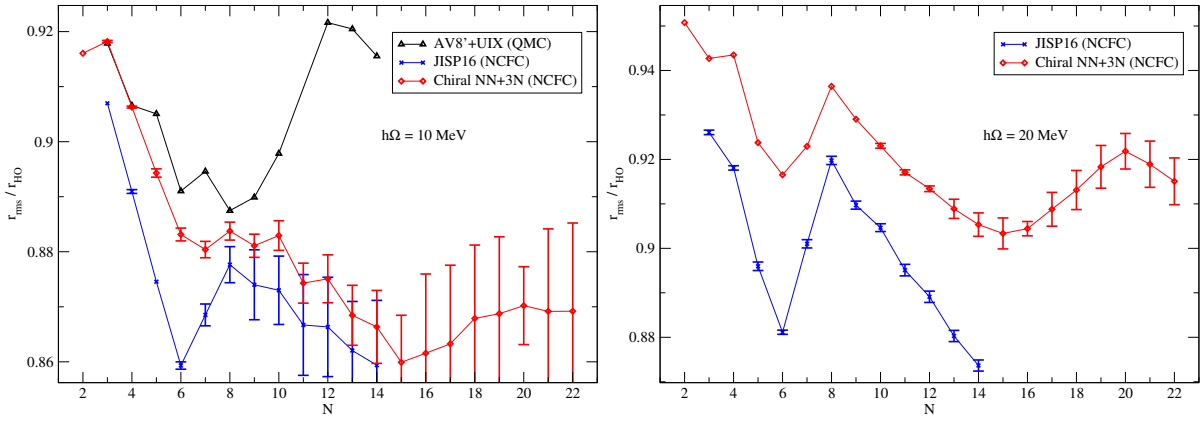

Figure 5.9 (Color online.) Single-particle rms radii for the lowest energy states of $N = 2 - 22$ systems in 10 MeV and 20 MeV HO traps for various interactions as indicated in the legend, scaled by the values expected from noninteracting neutrons in a pure harmonic potential. The results for AV8′+UIX and JISP16 are taken from Ref. (4) (note that Fig. 9 of Ref. (4) shows the rms radii of the states with lowest J, which are not always the lowest energy states).

In Fig. 5.9, we show the rms radii scaled by the values of their non-interacting limits. The odd-even staggering in the 10 MeV trap is much clearer in this view. In the 20 MeV trap we see some slight staggering in JISP16 in the *sd* shell that wasn't easily discernable in the the unscaled figure. Also, in the 20 MeV trap, we see a pattern wherein the radii fall below their non-interacting limits in the middle of the *p* and *sd* shells but return closer to it for the shell closures.

We portray the internal energies (scaled) in Fig. 5.10. We define internal energy $E_{\rm int}$ as the total energy $E_{\rm tot}$ as plotted Fig. 5.3 minus the expectation value of the HO trap potential computed with the ground-state wavefunction:

$$E_{\rm int} = E_{\rm tot} - \frac{1}{2} M\Omega \sum_i r_i^2 = E_{\rm tot} - \frac{1}{2} M\Omega A r_{\rm rms}^2 \tag{5.4}$$

The internal energy is directly related to the ground-state energy and radius, so we calculate it directly from those two quantities instead of performing another extrapolation. We use the error propagation system discussed in section 5.4. In the 10 MeV trap we see strong odd-even staggering, both with the chiral results and with JISP16, which indicates that significant pairing effects are present in both these *ab initio* results. AV8′+UIX shows some pairing effects below $N = 9$, but very little for heavier neutron drops. AV8′+UIX and JISP16 are both near

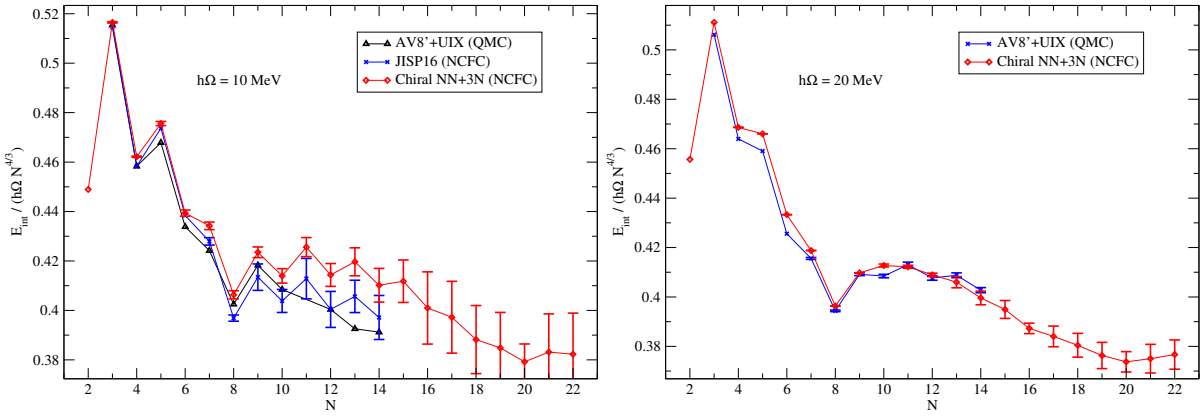

Figure 5.10 (Color online.) Internal energy (scaled) for the lowest-energy states of N-neutron systems in 10 MeV and 20 MeV HO traps as a function of N for various interactions as indicated in the legend. The results for AV8′+UIX and JISP16 are taken from Ref. (4).

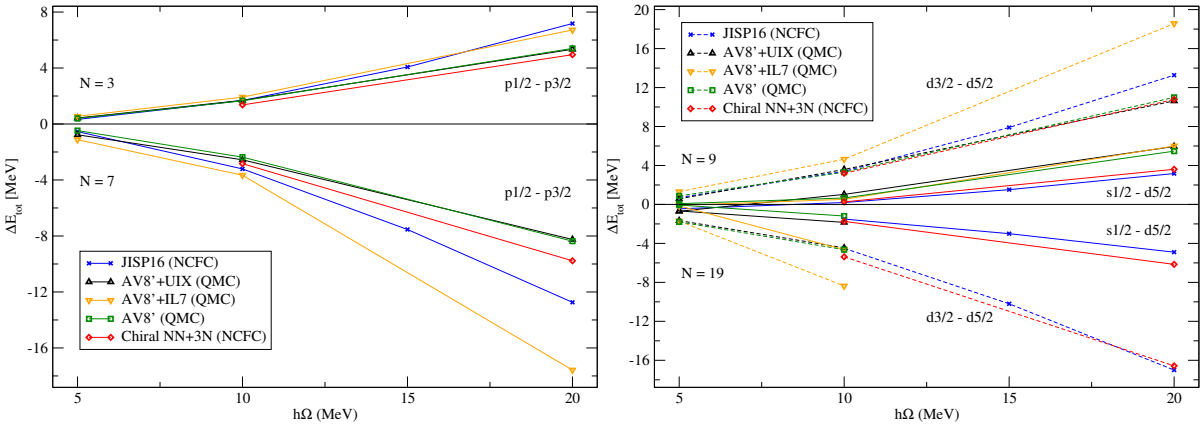

Figure 5.11 (Color online.) Energy splittings between levels of different J for various interactions as indicated in the legend. The results for AV8′+3N and JISP16 are taken from Ref. (4).

the chiral results until $N = 9$, after which they tend lower. In contrast, all JISP16 internal energies in the 20 MeV trap agree well with chiral predictions.

5.6 Level Splittings

We move now to energy level splittings for systems adjacent in N to closed energy shells. We can interpret these level splittings in the context of a simple phenomenological mean-field framework, wherein we model the inter-nucleon interactions as one mean field, generated by and affecting all the nucleons, effectively reducing an A-body problem to A one-body problems.

A common phenomenological choice for this mean field is the Woods-Saxon phenomenological potential. We base our discussion here on Ref. (66). The Woods-Saxon potential v_{WS} is given by

$$v_{\mathrm{WS}} = -V_0 f(r) \tag{5.5}$$

where f determines the radial structure of the potential:

$$f(r) = \frac{1}{1 + e^{(r-R)/a}} \tag{5.6}$$

V_0 is the depth of the potential well, which depends on the proton and neutron counts. a is a measure of the slope of the surface, taken to be 0.67 fm as in Ref. (66), and R is the radius of the nuclear system, defined in terms of nucleon count A and base radius r_0:

$$R = r_0 A^{1/3} = (1.27 \text{ fm}) A^{1/3} \tag{5.7}$$

v_{WS} is the mean field potential generated by the system's nucleons, so we can treat f as a description of nucleon density. The level splittings arise from the spin-orbit interaction between nucleon orbital and spin angular momentum. This interaction depends on the relative direction of $\boldsymbol{L}$ and $\boldsymbol{S}$, which is reflected in the coupled total angular momentum j; we expect, then, to see nucleons split into subshells depending on their j values.

Following Refs. (66) and (67), we model the spin-orbit interaction by adding a term v_{LS} proportional to $\boldsymbol{L} \cdot \boldsymbol{S}$:

$$v_{\mathrm{LS}}(r) = v_{\mathrm{LS}}^{(0)} \left(\frac{r_0}{\hbar}\right)^2 \frac{1}{r} \left[\frac{d}{dr} f(r)\right] \boldsymbol{L} \cdot \boldsymbol{S} \tag{5.8}$$

where $V_{\mathrm{LS}}^{(0)}$ is taken to be $v_{\mathrm{LS}}^{(0)} = 0.44 V_0$. $\boldsymbol{L} \cdot \boldsymbol{S}$ is positive when $\boldsymbol{L}$ and $\boldsymbol{S}$ are parallel and negative when they are antiparallel. For a given orbital angular momentum l, $\boldsymbol{L}$ and $\boldsymbol{S}$ are parallel at the higher j value; with the negative sign in Equation 5.8 for v_{LS}, then, we would expect the spin-orbit splitting to shift the higher-j state within the l-division of an energy shell downward.

The spin-orbit term in Equation 5.8 is phenomenological in origin; the source of nuclear spin-orbit splitting is not well-understood. *ab initio* level splitting results therefore offer a valuable comparison and avenue for further inquiry. In addition, we can use the level splittings to investigate subshell structure, comparing with our insights from the energy differences as discussed in section 5.4.

We investigate here the level splittings for systems adjacent in N to closed energy shells: $N = 3$ and 7 in the p-shell and $N = 9$ and 19 in the sd-shell. For these systems, we can interpret the level splittings as an indication of the relative energies of the various subshells. At $N = 9$, for example, we have one valence nucleon in the sd-shell. In its ground state, it will go into the energetically favourable $d_{5/2}$ shell, and in its sd-shell excited states, it can go in the $s_{1/2}$ or $d_{3/2}$ subshells; comparing those energy levels, then, allows us to compare the subshell energies. At $N = 19$, instead of one neutron we have one "hole" in the sd-shell, and our level splittings are negative to indicate removal from instead of addition to the subshells in question. We therefore observe predominantly negative splittings for the "one-hole" systems $N = 7$ and 19 in Fig. 5.11.

In Fig. 5.11 we present the p- and sd-shell level splittings plotted against the HO trap strength. We use the energy extrapolation discussed in section 5.3, and propagate error as discussed in section 5.4. As shown in Fig. 5.11, the splittings in our chiral results and comparison data increase uniformly with trap strength in both the p- and sd-shells. We can interpret this through the phenomenological model with Equation 5.8. As the strength of the external harmonic potential that we impose on neutron drops increases from 10 MeV to 20 MeV, we expect the radial structure to become more compact. In the phenomenological model, as the nucleon density increases at low r, the $1/r$ term in v_{LS} becomes less suppressive. Similarly, as the density becomes more localized, the gradient, appearing above as the derivative of f, becomes larger. In this simple phenomenological model, then, we would expect a more compact nucleon structure to produce a stronger spin-orbit splitting term. The increase in level splitting from the 10 MeV trap to the 20 MeV trap therefore matches our expectations. This also largely matches our observations in section 5.4, where we note that many of the subshell closures are stronger at 20 MeV than at 10 MeV.

In the sd-shell, we see a stronger splitting at $d_{3/2} - d_{5/2}$ than at $s_{1/2} - d_{5/2}$ both at $N = 9$ and $N = 19$ for all interactions. This is another indication of the shell closure we see at $N = 16$ in the energy differences in section 5.4; the $d_{5/2}$ and $s_{1/2}$ subshells are both of lower energy than the $d_{3/2}$ subshell, so they are filled first. Their total multiplicity is $6 + 2 = 8$, so, together

with the 8 neutrons in the full s and p shells, they hold 16 neutrons. The next neutron added, then, must bridge the $d_{3/2} - d_{5/2}$ energy gap, and we have a subshell closure.

We see a good agreement between all interactions at $N = 3$ in the p-shell. At $N = 7$, AV8′+IL7 has a stronger splitting, and the chiral results are close to AV8′and AV8′+IL7. AV8′+IL7 has a stronger $d_{3/2} - d_{5/2}$ splitting at $N = 9$, and at $N = 19$ through trap strength 10 MeV. The chiral results are close to JISP16 for $s_{1/2} - d_{5/2}$ at $N = 9$ and $d_{3/2} - d_{5/2}$ at $N = 19$, but match AV8′and AV8′+UIX more closely for $d_{3/2} - d_{5/2}$ at $N = 9$.

CHAPTER 6. CALCIUM 48

6.1 Overview

The neutron drop results discussed in chapter 5 illuminate important differences in the $T = 3/2$ channel for the various inter-nucleon potentials that we investigate, but none of these results is directly open to experiment. A recent theoretical effort (6) has uncovered a strong correlation between neutron drop radii and the experimentally observable neutron skin thickness in ^{48}Ca and ^{208}Pb; such a correlation provides an experimental window through which to investigate neutron drops, and, consequently, the behavior of various inter-nucleon potentials in the $T = 3/2$ channel.

In this chapter, we present several additional $N = 20$ neutron drop radii calculations; we use the correlation from Ref. (6) to calculate the corresponding ^{48}Ca neutron skin depths, and compare them with an experimentally determined range.

In addition to the importance of this correlation, ^{48}Ca is traditionally considered to be doubly magic; that is, its protons and neutrons all fall into closed energy shells and subshells. In part because of the resulting stability, ^{48}Ca cannot undergo a single $n \rightarrow p + e^- + \nu$ beta decay reaction to ^{48}Sc. It can, however, undergo a double beta decay reaction to ^{48}Ti in which two beta decays occur simultaneously. This makes ^{48}Ca a candidate for neutrinoless double beta decay, where one virtual neutrino is exchanged between beta decay reactions; since observation of this effect would mean that a neutrino is its own antiparticle, ^{48}Ca is of considerable theoretical and experimental interest in its own right. (68)

Table 6.1 Chiral NCFC $N = 20$, 10 MeV trap neutron drop radii results with varying SRG flow parameter α and UV regulator cutoff Λ. Numerical uncertainties are estimated as discussed in Section 5.5 and included in parentheses for the last digit.

α (fm^4)	Λ (MeV/c)	r_{rms} (fm)
0.04	400	3.12(5)
0.08	400	3.11(4)
0.04	500	3.08(7)
0.08	500	3.07(2)

6.2 Neutron Skin Depth and Neutron Drop Radius

Ref. (6) investigates the connection between the neutron skin depth of ^{48}Ca and the radius of an $N = 20$ neutron drop in a 10 MeV harmonic trap calculated with the same inter-nucleon potential. The neutron skin depth Δr_{np} is defined as the rms radius for protons subtracted from the rms radius for neutrons; because it is accessible to experiment, in combination with the correlation to neutron drop radius it can serve as an experimental access point to neutron drop systems.

Ref. (6) performs these calculation within the framework of nuclear density functional theory (DFT). Nuclear DFT assumes that the energy of a many-nucleon system depends only on one overall density field for protons and one for neutrons. Each nucleon contributes to its respective density field, but the density fields do not contain enough information to recover those individual contributions. In an $(N + Z)$-body system, the total volume-integral of the neutron field is N and the total volume-integral of the proton field is Z. An "energy density functional" (EDF) maps these two spatial densities to an energy; it is a "functional" because it maps a (spatial in this case) function to a value. Each EDF encodes a particular choice of inter-nucleon potential. For a given choice of EDF and system, then, the ground state nucleon densities of that system are the densities that minimize the EDF. Once these ground state densities are calculated, other one-body properties like the rms radius and neutron skin depth can be recovered from them.

A wide range of nuclear EDFs, including 13 relativistic EDFs and 12 non-relativistic EDFs, appear in Fig. 6.1, adapted from Ref. (6), where neutron skin thickness is plotted against the radius of the $N = 20$ neutron drop system in a 10 MeV harmonic trap. The 25 EDFs

investigated have a high variance in neutron skin depths and neutron drop radii, but are all tightly related by a strong linear correlation. This remarkable relation can serve as a starting point for experimental probes of neutron drop systems and the $T = 3/2$ channel of inter-nucleon interactions. As Ref. (6) discusses, this correlation is not unique to the $N = 20$ neutron drop system or the 10 MeV external trap; the correlation holds as long as the central density of the neutron drop remains at approximately the saturation density of neutrons in the centers of heavy nuclei. The external trap strength, then, can be adjusted simultaneously with the number of neutrons to extend this correlation to other neutron drop systems.

Fig. 6.1 also includes an inferred experimental value for the neutron skin thickness of ^{48}Ca, shown in green on the vertical axis. ^{48}Ca is the subject of much current experimental interest, but its neutron skin thickness is not yet directly experimentally determined; Ref. (6) uses a prediction for ^{48}Ca skin depth that relies on experimental results for the electric dipole polarizability of other nuclei, as discussed in Ref. (69).

Using the linear correlation, Ref. (6) "reflects" the experimental ^{48}Ca neutron skin thickness value onto the horizontal axis to find a range of $N = 20$ neutron drop radii that matches the experimental skin thickness data. This allows us to perform meaningful comparisons between calculated neutron drop radii and the neutron drop radii suggested by experiment. From Fig. 6.1, it appears that the relativistic DFT results are farther outside the experimental range, while many of the nonrelativistic DFT results are close. In particular, SKP, SLy4, SLy5, and SLY6 are within the experimental range; SKM* is within the experimental range for the skin depth, but not for the $N = 20$ radius.

To include our results in Fig. 6.1, we extend our 10 MeV trap, $N = 20$ neutron drop calculations, discussed in Section 5.5, to a wider range of interaction parameters. We vary the SRG flow parameter, investigating $\alpha = 0.04$ fm^4 and 0.08 fm^4, and the UV cutoff, investigating $\Lambda = 400$ MeV/c and 500 MeV/c. The combination of these two parameters gives us four different interactions; we present the results of these calculations in Table 6.1, with uncertainties estimated as in Section 5.5. In Fig. 6.1, we plot the resulting four ranges of neutron drop radii on the horizontal axis. With the help of the correlation, we have determined the corresponding "reflected" ^{48}Ca skin depths and plotted them on the vertical axis.

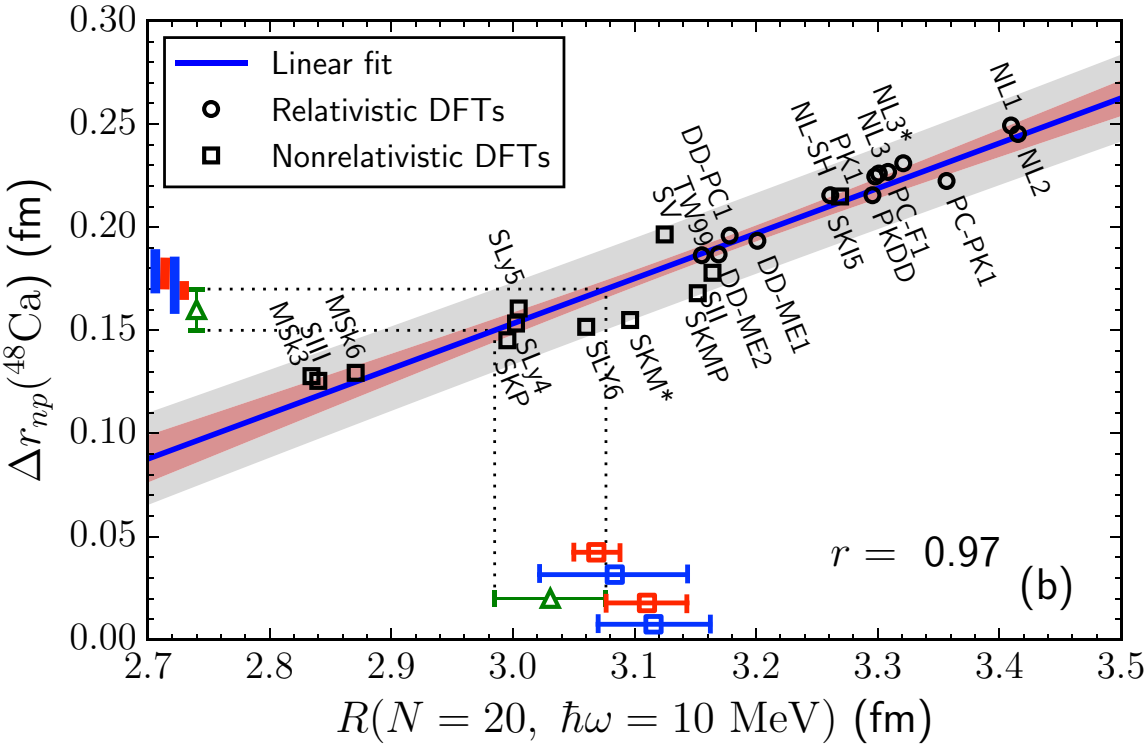

Figure 6.1 (Color online.) Correlation between $N = 20$ neutron drop radii in a 10 MeV trap and ^{48}Ca neutron skin thicknesses for a variety of interactions. Our chiral NCFC neutron drop results with different choices of UV regulator cutoffs and SRG flow parameters (see the text) are shown in red and blue on the horizontal axis. We use the linear correlation to predict chiral NCFC ^{48}Ca results, which are shown on the vertical axis. The remainder of the figure is reproduced from Ref. (6). The outer gray region represents a 95% prediction interval; we would expect new data points to be within that region 95% of the time. The inner red region represents a 95% confidence interval. That is, if we assume a linear parameterization for the correlation, every time we perform an independent estimation of the parameterization, we are 95% likely to produce parameters that include that region. The green intervals indicate an experimental range of ^{48}Ca neutron skin depth values, calculated from an experimental electric dipole polarizability result (69), and the corresponding neutron drop radius predicted from the correlation.

Our chiral results agree reasonably well with the result inferred from experiment, though on average they are somewhat higher. The two $\Lambda = 500$ MeV/c results are just above the experimental error range and have error ranges that clearly overlap it; the $\Lambda = 400$ MeV/c results are too high (with a very slight overlap for $\alpha = 0.04$ fm^4). As we would expect from the lack of dependence on α, discussed in Section 5.2, the results at each value of α are close together. We note that our results are closer to the experimental result than all of the relativistic DFT results in Fig. 6.1.

We have performed ^{48}Ca calculations for the chiral interactions discussed above, but the size of the problem limited us to $N_{\max} = 2$, and we were unable to produce meaningfully converged results. The neutron skin thickness that we calculated was approximately 0.16 fm to 0.21 fm, and did not depend significantly on the choice of chiral interaction, but due to the lack of convergence we do not regard that as a reliable result.

As discussed above, the linear correlation can be extended to any neutron drop system where the external harmonic potential is calibrated to produce a central density approximately equal to the saturation density of neutrons in the centers of heavy nuclei. Fig. 6.2, adapted from Ref. (6), shows the experimentally predicted range of radii for three neutron drop systems compared with several *ab initio* results. Ref. (6) demonstrates a correlation for ^{208}Pb as well as ^{48}Ca; Fig. 6.2 shows the range of neutron drop radii predicted by three separate ^{208}Pb experimental results and the one ^{48}Ca result discussed above. The chiral results that we include in Fig. 6.2 are calculated at SRG parameter $\alpha = 0.08$ fm^4 and UV cutoff $\Lambda = 500$ MeV/c, as in Section 5.5.

The results in Fig. 6.2 are similar in some respects to the results presented in Chapter 5; chiral results are close to AV8′ and JISP16, and AV8′+UIX is consistently underbound. The experimental ranges suggest that the chiral interaction is also slightly underbound. The chiral radius at $N = 20$ is inside all four experimental ranges, though it occurs at the highest part of three of them. At $N = 14$, the chiral result is above all the experimental ranges except the range predicted by the ^{208}Pb antiprotonic atom measurement, which it is barely inside. The well-converged chiral result at $N = 8$ is at the upper edge of two of the ^{208}Pb experimental

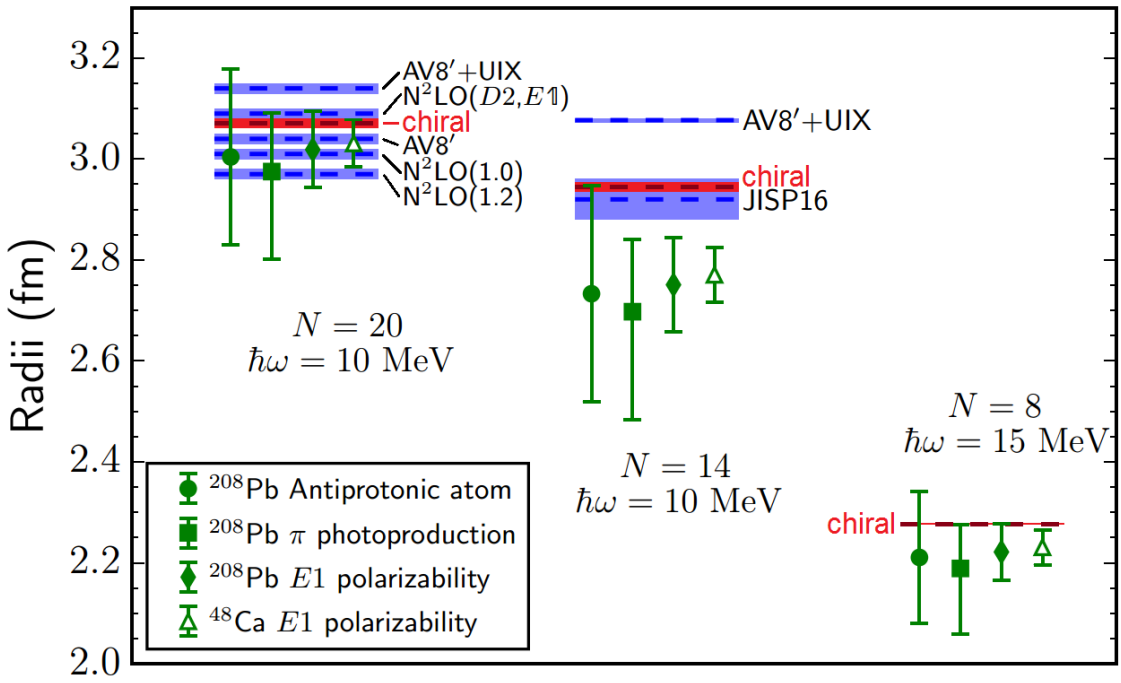

Figure 6.2 (Color online.) Selection of neutron drop radii results for $N = 20$ and 14 systems in 10 MeV traps and the $N = 8$ system in a 15 MeV trap. *Ab initio* results are shown as solid regions; our chiral NCFC results are in red, and results reproduced from Ref. (6) are in blue. Experimental results are shown in green; the experimental ranges are determined from ^{48}Ca and ^{208}Pb neutron skin depth experimental results (discussed in more depth in Ref. (6)) via the linear correlation discussed in Section 6.2. Figure adapted from Ref. (6).

ranges, and slightly above the ^{48}Ca experimental range; as with the other systems, it is inside the ^{208}Pb antiprotonic atom measurement.

This theoretical link between exotic neutron-only systems and neutron skin depth offers an important opportunity to compare our work to experiment. We look forward to more precise ^{48}Ca experimental results and additional chiral NCFC neutron drop comparisons. Chiral NCFC ^{48}Ca results would be ideal to check our agreement with the correlation, but we must wait for new *ab initio* methods or substantially increased computing power before such calculations are feasible. In the meantime, we have used the correlation to compare our neutron drop results with experiment, demonstrating an agreement that is comparable or superior to many of the DFT results discussed in Ref. (6).

CHAPTER 7. CONCLUSION

We have presented a detailed *ab initio* strategy for nuclear structure calculations and a wide range of neutron drop calculations using our formalism. We have shown neutron drop results in 10 and 20 MeV external harmonic traps, with neutron counts ranging from 2 – 40. Our results include total binding energies, single and double energy differences, internal energies, radii, and level splittings. For comparison, we have presented GFMC and AFDMC results using AV8′+3N interactions, and additional NCFC results using the JISP16 interaction. We have found significant dependences on the selected Hamiltonians, which should inform the phenomenological energy-density functionals that may be derived from them.

We have adopted chiral NN and chiral 3N interactions up to N^3LO for NN and up to N^2LO for 3N. Future work will extend these calculations to next-generation chiral Hamiltonians and investigate the sensitivity of observables to the underlying chiral Hamiltonian. Furthermore, comparison of our *ab initio* results to predictions from state-of-the-art energy-density functionals will help constrain the latter in regimes of extreme isospin.

Building on our neutron drop radii calculations, we have presented a wider range of $N = 20$ results in the context of the recent study linking neutron drop radii to neutron skin depth in ^{48}Ca and ^{208}Pb. With the help of that correlation, we have compared our results to experiment and demonstrated agreement that compares favorably with many DFT results. This study and its correlation serve as a valuable theoretical link to help facilitate comparisons of neutron drop properties with experiment.

In addition, we have presented our strategy for accelerating our code on the GPU. The highly non-regular angular momentum structure of our input matrices and decoupling code presents a unique challenge in the grid-based world of GPU computing. We have managed to demonstrate a 2.2x – 2.7x speedup in the accelerated portion of the code and a 1.2x – 1.4x

speedup in the entire code. In our approach we investigated several possibilities for optimization, but left several others for later attempts. In the future, we are interested to see how our acceleration strategy can be streamlined and improved by investigating different levels of GPU parallelization or calling on other techniques.

One possible direction of attack is a GPU-accelerated "on-the-fly" version of MFDn. At present, the GPU-accelerated MFDn code generates the many-body Hamiltonian matrix once and stores it on-core throughout the entire diagonalization process. Another option would be to generate sections of the many-body Hamiltonian every Lanczos iteration as the diagonalization routine requires them. Our code is strongly memory-bound, so this would vastly extend the range of problems we could address. At present, however, such an approach takes a prohibitive amount of time. We are cautiously interested in the possibility that a more streamlined GPU acceleration strategy could reduce the time cost to the point where such calculations are more feasible.

We look forward to new physics opportunities as we explore GPU acceleration and other multiprocessing paradigms for tomorrow's fastest supercomputers.

APPENDIX. THE COUPLED-JT BASIS

1 The Coupled-JT Basis

We present here a more rigorous description of the coupled-JT format discussed in chapter 4. We write a 3-body MBS as $|n_a l_a j_a m_a t_a\rangle|n_b l_b j_b m_b t_b\rangle|n_c l_c j_c m_c t_c\rangle$, and denote $n_a l_a j_a m_a t_a$ as a and $n_a l_a j_a$ as $\bar{a}$ so the m-scheme state can be shortened to $|abc\rangle$. We rewrite it coupling j_a with j_b into J_{ab}:

$$\sum_{J_{ab}} |\bar{a}t_a; \bar{b}t_b; c; J_{ab}m_{ab}\rangle\langle J_{ab}m_{ab}|m_a m_b\rangle$$

where the Clebsch-Gordan coefficient (CGC) that couples j_a with j_b into J_{ab} with magnetic projection m_{ab} is written as $\langle J_{ab}m_{ab}|m_a m_b\rangle$. We then couple J_{ab} with j_c into $J_{(ab)c}$:

$$\sum_{J_{ab}} \sum_{J_{(ab)c}} |\bar{a}t_a; \bar{b}t_b; \bar{c}t_c; J_{ab}J_{(ab)c}m_{(ab)c}\rangle\langle J_{ab}m_{ab}|m_a m_b\rangle\langle J_{(ab)c}m_{(ab)c}|m_{ab}m_c\rangle$$

Finally, we couple isospin in the same fashion. As discussed in Section 2.2, we use T for isospin magnitude, and $T_a = T_b = T_c = 1/2$ is implied; as above, we use t for isospin projection.

$$\sum_{J_{ab}} \sum_{J_{(ab)c}} \sum_{T_{ab}} \sum_{T_{(ab)c}} |\bar{a}\bar{b}\bar{c}; J_{ab}J_{(ab)c}m_{(ab)c}; T_{ab}T_{(ab)c}t_{(ab)c}\rangle$$
$$\times \langle J_{ab}m_{ab}|m_a m_b\rangle\langle J_{(ab)c}m_{(ab)c}|m_{ab}m_c\rangle\langle T_{ab}t_{ab}|t_a t_b\rangle\langle T_{(ab)c}t_{(ab)c}|t_{ab}t_c\rangle$$

A coupled-JT state has definite n, l, and j for each of its three SPSs, but instead of three definite angular momentum projections, it has a definite total angular momentum sum for the first two SPSs and for all three SPSs together, and likewise for isospin. The ket $|\bar{a}\bar{b}\bar{c}; J_{ab}J_{(ab)c}m_{(ab)c}; T_{ab}T_{(ab)c}t_{(ab)c}\rangle$ above is thus a coupled-JT basis state.

2 Recovering m-scheme

We recover m-scheme elements by decoupling them one-at-a-time. Starting with the m-scheme state that we defined as a linear combination of coupled-JT states in the previous section, we take the inner product with another coupled-JT state to calculate the projector between the m-scheme and coupled-JT bases:

$$\sum_{J_{ab}} \sum_{J_{(ab)c}} \sum_{T_{ab}} \sum_{T_{(ab)c}} \langle \bar{x}\bar{y}\bar{z}; j_{xy} j_{(xy)z}; T_{xy} T_{(xy)z} | \bar{a}\bar{b}\bar{c}; J_{ab} J_{(ab)c}; T_{ab} T_{(ab)c} \rangle$$
$$\times \langle J_{ab} m_{ab} | m_a m_b \rangle \langle J_{(ab)c} m_{(ab)c} | m_{ab} m_c \rangle \langle T_{ab} t_{ab} | t_a t_b \rangle \langle T_{(ab)c} t_{(ab)c} | t_{ab} t_c \rangle$$

Using several orthogonality relations to simplify:

$$\delta_{ax}\delta_{by}\delta_{cz} \langle j_{xy} m_{ab} | m_a m_b \rangle \langle j_{(xy)z} m_{(ab)c} | m_{ab} m_c \rangle \langle T_{xy} t_{ab} | t_a t_b \rangle \langle T_{(xy)z} t_{(ab)c} | t_{ab} t_c \rangle$$

Here we use δ-functions to refer to n, l, and j indices only.

Now we can calculate an m-scheme matrix element $\langle abc|V|a'b'c' \rangle$. Inserting the identity:

$$\langle abc|A|a'b'c' \rangle = \sum \langle abc|\bar{x}\bar{y}\bar{z}; j_{xy} j_{(xy)z} T_{xy} T_{(xy)z} \rangle$$
$$\langle \bar{x}\bar{y}\bar{z}; j_{xy} j_{(xy)z} T_{xy} T_{(xy)z} | V | \bar{x'}\bar{y'}\bar{z'}; j_{x'y'} j_{(x'y')z'} T_{x'y'} T_{(x'y')z'} \rangle$$
$$\langle \bar{x'}\bar{y'}\bar{z'}; j_{x'y'} j_{(x'y')z'} T_{x'y'} T_{(x'y')z'} | a'b'c' \rangle$$

where the sum is over all coupled-JT states, both primed and unprimed. Using the projector we calculated:

$$\langle abc|A|a'b'c' \rangle = \sum \delta_{ax}\delta_{by}\delta_{cz}\delta_{a'x'}\delta_{b'y'}\delta_{c'z'}$$
$$\times \langle j_{xy} m_{ab} | m_a m_b \rangle \langle j_{(xy)z} m_{(ab)c} | m_{ab} m_c \rangle \langle T_{xy} t_{ab} | t_a t_b \rangle \langle T_{(xy)z} t_{(ab)c} | t_{ab} t_c \rangle$$
$$\times \langle j_{x'y'} m_{a'b'} | m_{a'} m_{b'} \rangle \langle j_{(x'y')z'} m_{(a'b')c'} | m_{a'b'} m_{c'} \rangle \langle T_{x'y'} t_{a'b'} | t_{a'} t_{b'} \rangle \langle T_{(x'y')z'} t_{(a'b')c'} | t_{a'b'} t_{c'} \rangle$$
$$\times \langle \bar{x}\bar{y}\bar{z}; j_{xy} j_{(xy)z} T_{xy} T_{(xy)z} | V | \bar{x'}\bar{y'}\bar{z'}; j_{x'y'} j_{(x'y')z'} T_{x'y'} T_{(x'y')z'} \rangle$$

The δ-functions restrict the calculation to a specific nlj block of both the input and the output matrices; the transformation from m-scheme to coupled-JT does not mix blocks. An m-scheme element, then, is the sum of all coupled-JT elements in the block, weighted by the eight CG-coefficients used to couple angular momentum and isospin.

Bibliography

[1] B. S. Pudliner, A. Smerzi, J. Carlson, V. R. Pandharipande, S. C. Pieper, et al., Phys. Rev. Lett. 76 (1996) 2416.

[2] S. K. Bogner, R. J. Furnstahl, H. Hergert, M. Kortelainen, P. Maris, et al., Phys. Rev. C 84 (2011) 044306.

[3] S. Gandolfi, J. Carlson, S. C. Pieper, Phys. Rev. Lett. 106 (2011) 012501.

[4] P. Maris, J. P. Vary, S. Gandolfi, J. Carlson, S. C. Pieper, Phys. Rev. C 87 (2013) 054318.

[5] H. D. Potter, S. Fischer, P. Maris, J. P. Vary, S. Binder, A. Calci, J. Langhammer, R. Roth, Phys. Lett. B. 739 (2014) 445.

[6] P. W. Zhao and S. Gandolfi, arXiv:1604.01490 (2016).

[7] P. Navrátil, J. P. Vary, B. R. Barrett, Phys. Rev. Lett. 84 (2000) 5728.

[8] P. Navrátil, J. P. Vary, B. R. Barrett, Phys. Rev. C 62 (2000) 054311.

[9] B. R. Barrett, P. Navrátil, J. P. Vary, Prog. Part. Nucl. Phys. 69 (2013) 131.

[10] P. Maris, J. P. Vary, A. M. Shirokov, Phys. Rev. C 79 (2009) 014308.

[11] P. Navrátil, S. Quaglioni, I. Stetcu, B. R. Barrett, J. Phys. G 36 (2009) 083101.

[12] P. Maris, A. M. Shirokov, J. P. Vary, Phys. Rev. C 81 (2010) 021301.

[13] R. Roth, Phys. Rev. C 79 (2009) 064324.

[14] R. Roth, J. Langhammer, A. Calci, S. Binder, P. Navrátil, Phys. Rev. Lett. 107 (2011) 072501.

[15] R. Roth, S. Binder, K. Vobig, A. Calci, J. Langhammer, P. Navrátil, Phys. Rev. Lett. 109 (2012) 052501.

[16] P. Maris, J. P. Vary, P. Navrátil, W. E. Ormand, H. Nam, D. J. Dean, Phys. Rev. Lett. 106 (2011) 202502.

[17] P. Navrátil, S. Quaglioni, Phys. Rev. Lett. 108 (2012) 042503.

[18] E. Epelbaum, H.-W. Hammer, U.-G. Meißner, Rev. Mod. Phys. 81 (2009) 1773.

[19] R. Machleidt, D. R. Entem, Phys. Rep. 503 (2011) 1.

[20] S. D. Głazek, K. G. Wilson, Phys. Rev. D 48 (1993) 5863.

[21] F. Wegner, Ann. Phys. 506 (1994) 77.

[22] S. K. Bogner, R. J. Furnstahl, P. Maris, R. J. Perry, A. Schwenk, J. P. Vary, Nucl. Phys. A 801 (2008) 21.

[23] H. Hergert, R. Roth, Phys. Rev. C 75 (2007) 051001.

[24] S. K. Bogner, R. J. Furnstahl, A. Schwenk, Prog. Part. Nucl. Phys. 65 (2010) 94.

[25] R. J. Furnstahl, Nucl. Phys. Proc. Suppl. 228 (2012) 139.

[26] E. D. Jurgenson, P. Navrátil, R. J. Furnstahl, Phys. Rev. Lett. 103 (2009) 082501.

[27] R. Roth, A. Calci, J. Langhammer, S. Binder, Phys. Rev. C 90 (2014) 024325.

[28] B. S. Pudliner, V. R. Pandharipande, J. Carlson, S. C. Pieper, R. B. Wiringa, Phys. Rev. C 56 (1997) 1720.

[29] S. C. Pieper, AIP Conf. Proc. 1011 (2008) 143.

[30] A. M. Shirokov, A. I. Mazur, S. A. Zaytsev, J. P. Vary, T. A. Weber, Phys. Rev. C 70 (2004) 044005.

[31] A. M. Shirokov, J. P. Vary, A. I. Mazur, T. A. Weber, Phys. Lett. B 644 (2007) 33.

[32] D. R. Entem, R. Machleidt, Phys. Rev. C 68 (2003) 041001.

[33] E. Epelbaum, A. Nogga, W. Glöckle, H. Kamada, U.-G. Meißner, H. Witała, Phys. Rev. C 66 (2002) 064001.

[34] P. Navrátil, Few Body Syst. 41 (2007) 117.

[35] D. Gazit, S. Quaglioni, P. Navrátil, Phys. Rev. Lett. 103 (2009) 102502.

[36] E. D. Jurgenson, P. Navrátil, R. J. Furnstahl, Phys. Rev. C 83 (2011) 034301.

[37] E. D. Jurgenson, P. Maris, R. J. Furnstahl, P. Navrátil, W. E. Ormand, et al., Phys. Rev. C 87 (2013) 054312.

[38] P. Maris, J. P. Vary, A. Calci, J. Langhammer, S. Binder and R. Roth, Phys. Rev. C 90 (2014) 014314.

[39] P. Sternberg, E. G. Ng, C. Yang, P. Maris, J. P. Vary, M. Sosonkina, H. V. Le, in: SC, IEEE/ACM, 2008, p. 15.

[40] P. Maris, M. Sosonkina, J. P. Vary, E. Ng, C. Yang, Procedia CS 1 (2010) 97.

[41] H. M. Aktulga, C. Yang, E. G. Ng, P. Maris, J. P. Vary, Topology-aware mappings for large-scale eigenvalue problems, in: C. Kaklamanis, T. Papatheodorou, P. G. Spirakis (Eds.), Euro-Par, Vol. 7484 of Lecture Notes in Computer Science, Springer, 2012, pp. 830–842.

[42] H. M. Aktulga, C. Yang, E. G. Ng, P. Maris, J. P. Vary, Concurrency Computat.: Pract. Exper. (2013), doi:10.1002/cpe.3129.

[43] P. Maris, H. M. Aktulga, M. A. Caprio, Ü. V. Çatalyürek, E. G. Ng, D. Oryspayev, H. Potter, E. Saule, M. Sosonkina, J. P. Vary, C. Yang, Z. Zhou, J. Phys. Conf. Ser. 403 (2012) 012019.

[44] J. P. Vary, 1992, The Many-Fermion-Dynamics Shell-model Code (Iowa State University) (unpublished).

[45] J. P. Vary and D. C. Zheng, 1994, The Many-FermionDynamics Shell-Model Code (Iowa State University) (unpublished).

[46] J. P. Vary, P. Maris, E. Ng, C. Yang and M. Sosonkina, "*Ab initio* nuclear structure – the large sparse matrix eigenvalue problem," J. Phys.: Conf. Series 180, 012083 (2009).

[47] D. H. Gloekner and R. D. Lawson, Phys. Lett. 53 (1974) 313.

[48] C. Lanczos, J. Res. Natl. Bur. Standards, Sect. B 45 (1950) 255.

[49] Y. Saad, "Numerical Methods for Large Eigenvalue Problems: Revised Edition", Society for Industrial and Applied Mathematics, Philadelphia (2011).

[50] NVIDIA, *CUDA C Programming Guide*. http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf, 2015. Accessed July 2016.

[51] D. Oryspayev, H. Potter, P. Maris, M. Sosonkina, J. P. Vary, S. Binder, A. Calci, J. Langhammer, R. Roth, IEEE 27th Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW) (2013) 1365.

[52] H. Potter et. al., Proc. "Nuclear Theory in the Supercomputing Era - 2013" (NTSE-2013), Ames, IA, USA, 13-17 May, 2013. 2014, p. 263.

[53] Oakridge Leadership Computing Facility, *CUDA Proxy: Managing GPU context*. https://www.olcf.ornl.gov/tutorials/cuda-proxy-managing-gpu-context, 2015. Accessed July 2016

[54] M. Włoch, D. J. Dean, J. R. Gour, M. Hjorth-Jensen, K. Kowalski, T. Papenbrock, P. Piecuch, Phys. Rev. Lett. 94 (2005) 212501.

[55] G. Hagen, T. Papenbrock, D. J. Dean, M. Hjorth-Jensen, Phys. Rev. Lett. 101 (2008) 092502.

[56] G. Hagen, M. Hjorth-Jensen, G. R. Jansen, R. Machleidt, T. Papenbrock, Phys. Rev. Lett. 108 (2012) 242501.

[57] G. Hagen, T. Papenbrock, M. Hjorth-Jensen, Rep. Prog. Phys. 77 (2014) 096302.

[58] G. Hagen, T. Papenbrock, D. J. Dean, M. Hjorth-Jensen, Phys. Rev. C82 (2010) 034330.

[59] A. G. Taube, R. J. Bartlett, J. Chem. Phys. 128 (2008) 044110.

[60] C. Forssén, J. P. Vary, E. Caurier, and P. Navrátil, Phys. Rev. C 77 (2008) 024301.

[61] P. Maris and J. P. Vary, "Ab initio Nuclear Structure Calculations of p-shell Nuclei with JISP16," invited mini-review paper, Int. J. Mod. Phys. E 22, 1330016 (2013); DOI: 10.1142/S0218301313300166.

[62] Andrey M. Shirokov, Vasily A. Kulikov, Pieter Maris, and James P. Vary, "Bindings and spectra of light nuclei with JISP16," in Nucleon-Nucleon and Three-Nucleon Interactions, ed. L. D. Blokhintsev and I. I. Strakovsky, Nova Science (2014), Chapter 8, p. 231; ISBN: 978-1-63321-053-0.

[63] C. Cockrell, J. P. Vary, and P. Maris, Phys. Rev. C 86 (2012) 034325.

[64] M. A. Caprio, P. Maris, J. P. Vary, Phys. Rev. C 86 (2012) 034312.

[65] M. A. Caprio, P. Maris, J. P. Vary, Phys. Rev. C 90 (2014) 034305.

[66] J. Suhonen, "From Nucleons to Nucleus", Springer-Verlag, Berlin, Heidelberg (2007).

[67] A. Bohr, B. R. Mottelson, "Nuclear Structure, vol 1", Benjamin, New York (1969).

[68] P. Vogel, "Double Beta Decay: Theory, Experiment, and Implications", in "Current Aspects of Neutrino Physics", ed. D. O. Caldwell, Springer-Verlag, Berlin, Heidelberg (2001).

[69] X. Roca-Maza, X. Viñas, M. Centelles, B. K. Agrawal, G. Colò, N. Paar, J. Piekarewicz, D. Vretenar, Phys. Rev. C 92 (2015) 064304.